DEVELOPMENT OF A VISION GUIDED VEHICLE

A thesis submitted
in partial fulfilment of the requirements
for the degree of

MASTER OF TECHNOLOGY

by
VADDI DURGA PRASAD

to the

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY-KANPUR
April-1994

ME-1994-M-PRA-DEV

CERTIFICATE

It is certified that the work contained in the thesis entitled *Development o* *Vision Guided Vehicle* by **Vaddi Durga Prasad**, has been carried out under supervision and that this work has not been submitted else where for a degree.

(Dr. Sudipto Mukherjee)

Assistant Professor
Dept. of Mechanical Engg.
Indian Institute of technology
Kanpur.

Karnick

(Dr. H. Karnick)

Associate Professor
Dept. of Computer Science & Eng
Indian Institute of technology
Kanpur.

Dedicated to

The Supreme Personality of Godhead

Sri Krishna

and to

my beloved parents

Sri. Vaddi Venkata Nageswara Rao

Smt. Nageswaramma

Acknowledgements

I would like to take this opportunity to extend my hearty thanks to my thesis supervisors **Dr. H. Karnick** and **Dr. S. Mukherjee** for giving me valuable guidance, freedom to experiment and providing access to facilities at Center for Robotics and Vision Lab.

It is my pleasure to associate with the Center for Robotics for nearly two years. I extend my sincere thanks to Mr. Susmit Sen who helped me a lot for succesfully completing my thesis work and in clearly understanding the concept of real time control. It is my pleasure to mention hearty thanks to Mr. Neel Mathews, Mrs. Anjali Kulkarni and Mr. K.K.Pathak who made my stay at I.I.T kanpur a memorable one. I am also very much thankful to Mr. Khurshid Alam, Mr. Soumya Bhattacharya and Mr. Anirvan Das Gupta for their help in carrying my thesis work.

I wish to thank Mr. R.M. Jha, Prem Prakash, Sharma, Namdev and others in manufacturing Science Lab who helped me in realizing the Mobile Platform.

I would like to thank my friends N. Srinivas, S.P. Chandra Shekhar, U.V. Sarma, Mahesh, I. Sreedhar and others for their help in documenting my thesis.

ABSTRACT

A Vision guide vehicle called, the Moving Platform has been implemented in the present work. The activities of a mobile robot such as perception, planning and control, and actuation are to some extent implemented on this Platform.

The hardware of the Mobile Platform consists of two rear wheels powered by D.C motors and a front passive wheel (caster). These motors are connected to a driver board, an electronic circuit for controlling motors, which is in turn connected to a P.C through its printer port. A program has been developed for controlling the two motors independently.

A path planning program implemented for a rectangular workspace with two polygonal obstacles. Given the start position of the Platform and required goal position this program gives an allowable path containing a set of straight lines.

Feed back of configuration of the Platform, in real time, is established by taking images of workspace by a C.C.D. camera, connected to a second P.C. Serial Port communication has been used for transferring data between computers. A program has been implemented to establish the feed back and to transfer the data through the serial port.

A program which integrates all the above mentioned programs together has been implemented successfully, leading to a demonstrable model of a vision guided system.

Contents

List of Figures

List of Tables

Nomenclature

H/W	Hardware
S/W	Software
C.C.D	Charge Coupled Device
PC1	PC-AT 80386
PC2	PC-AT 80486
x	Reference frame x-axis
y	Reference frame y-axis
X	Robot centered frame X-axis
Y	Robot centered frame Y-axis
FRC	Flight refueling corporation
Vehicle	Moving Platform
D.C	Direct current
A.C	Alternate current
A/D	Analog to digital
D/A	Digital to analog
T.V	Television
F.G	Fore ground
B.G	Back ground
P.C	Computer
Motor-1	Left side motor of the Moving Platform, when the front end of the Platform is towards the caster.
Motor-2	Right side motor of the Moving Platform, when the front end of the Platform is towards the caster.
Encoder-1	Optical encoder of the Motor-1
Encoder-1	Optical encoder of the Motor-1

Chapter 1

Introduction

1.1 Introduction

A robot can manipulate only those objects it can reach. Most industrial robots are fixed in place. Their workspace is limited by maximum extent of their linkages. Components are brought to the proximity of the robot and moved away by conveyors and other mechanical feed devices. To overcome the problems caused by the limited reach of robot arms, two approaches are subject of investigations. One is the flexible manufacturing cell, where the robot is fixed in place and the machines that it services are placed around it. As the robot can reach several machines it can service one machine while the others are performing their task. A second approach is to make the robot mobile. Muir and Newman defined a wheeled mobile robot as,

> A robot capable of locomotion on a surface solely through the actuation of wheel assemblies mounted on the robot and in contact with the surface. A wheel assembly is a device which allows or provides the relative motion between its mount and a surface on which it is intended to have a single Point or Line of rolling contact.

Research on mobile robots began in the late sixties with the Stanford Research Institute's pioneering work. Mobile robots were, and still are, a very popular and rich field for research in Artificial Intelligence applied to Robotics. Many researchers have investigated representations for world modeling , issues in planning and problem solving. Concomitantly, the complexity of the problems in sensing, perception and control were vastly under estimated. As a result the systems built did not perform as well as expected. The 1980's marked a strong resurgence of interest in mobile robots,

motivated by advances in processing power and sensor systems, and sensing technologies to the point that practical applications in restricted domains such as teleoperation etc. become feasible.

The major applications of mobile robots are:

1. Factory automation projects, where robotic vehicles are used to transport components between distant machining and assembly sites.
2. Operations in hazardous environments.
3. Planetary and space exploration, using autonomous rovers and probes.
4. The use of mobile robots for deep sea surveying and prospecting.
5. Aids for handicapped.
6. Military applications, such as "Battle field of the future", scenarios.

1.2 Capabilities Required for Autonomous Mobile Robots

Using various sensors, a mobile robot needs to acquire and manipulate a substantial model of its operational environment by extracting and interpreting information from the real world. Individual sensors are subjected to intrinsic limitations that can be over come only by utilizing multisensory systems. Additionally, mobile robots travel over extensive areas and must combine views obtained from many different locations into a single consistent world model, reflecting information acquired. Consequently, sensor interpretation and world modeling process must incorporate information supplied by functionally different sensors, cope gracefully with sensor errors and noise, and address uncertainty arising from imprecise knowledge of the robots location over time. The resulting world model will be used as the basis for crucial operations, including path planning and obstacle avoidance, position and motion estimation , navigation and landmark identification.

As a whole the activities of a mobile robot can be classified into three major conceptual areas.

1. Perception

   Perception encompasses tasks including sensor interpretation, sensor integration real-world modeling and recognition.

2. Planning and Control
   This includes task level planning, scheduling, and execution monitoring of overall robotic activity
3. Actuation
   Actuation comprises of navigational activities, detailed motion and action planning and actuator control.

A path planner specifies a motion to be executed by a moving platform, typically a sequence of connected lines, as explained earlier. The basic task of the real time controller is to make the platform execute the motion plan, e.g to follow generated path. Let us first assume that the motion plan is a free path $\mathcal{T}$. The task of the controller is to transform $\mathcal{T}$ into forces or torques to be exerted by the platform actuators (D.C motors). Typically this transformation is broken down into two steps. The first step, called trajectory generation, consists of deciding on the velocity profile along the path. It can be done prior to motion execution. The second step, called trajectory tracking , consists of computing the forces to be exerted by the actuators at each time in order to perform the desired motion. The trajectory tracking step may use the dynamic equation of the platform - the equation expressing the forces applied by the actuators equal the resultant of the various forces acting on the platform during motion, i.e the gravitational forces, frictional forces , centrifugal forces , etc.- to compute the force that has to be delivered by each actuator. If the dynamic equation used by the controller is a perfect model no feedback would be needed. However due to various disturbances, sensing (feedback) is necessary to determine the deviation between the desired state and actual state of the platform. While the motion is being executed the controller computes the actuator forces which tend to eliminate this deviation at some update rate. Figure 1.3 shows the relation between the path planner, the trajectory generator, the controller and the platform.

These various conceptual tasks clearly interact with and depend upon each other. In fact, experience with real systems reveal that complex inter connections and inter dependencies exist between the various sub systems with multiple flow channels for control and data.

1.3 A Brief Historical Review

Mobile robot research addresses many problems, including sensor interpretation and integration, real world modeling, actuator and sensor control, path planning and navigation, task level planning and plan execution, and the global monitoring and control of robotic systems as a whole. The following are the research projects conducted in the mobile robot domain.

- Shakey

  This is one of first mobile robots, developed at the Stanford Research Institute, Stanford university, California. Shakey had a T.V camera, an optical range finder and several touch sensors. It was operated in a highly constrained artificial environment, populated with large blocks and used a precompiled map of its surroundings. The project led to significant developments in logic based planning and problem solving techniques, but did not focuses on sensing or real world modeling issues, there by producing a system with extremely limited performance.

- Jason

  Developed at the university of California, Berkeley. Jason had an on-board manipulator as well as ultrasonic range sensors and infrared proximity detectors. Experimentation with jason exposed a variety of problems on teleoperation.

- The JPL Rover

  This JPL Rover has been developed by NASA's Jet Propulsion Laboratory, Pasadena, California, as a part of on going research program to develop planetary explorers. It was a semi autonomous vehicle equipped with a Laser range finder, a pair of T.V. camera's and a manipulator arm.

- The Stanford Cart

  Built at Stanford A.I laboratory, Stanford University, California.This was a simple remotely controlled mobile platform equipped with a camera mounted on a slider mechanism and a radio transmitter.

- The Hermies Robots

  The Hermies, a series of robots, developed at the Oak ridge National Laboratory, Tennessee incorporated sonar and vision sensors. Hermies have manipulators mounted on the mobile platform and on-board processing. They have been

used for research in visual perception, goal recognition, navigation in unknown dynamic environments and the development of task oriented manipulator strategies.

- MIT robots
  Several different vehicles have been developed at MIT's Artificial Intelligence laboratory, Cambridge, Massachusetts, to investigate robust low level sensing and control mechanisms, their use in layered control architecture.

- The Stanford Mobi
  An omnidirectional platform equipped with a stereo camera system and sonar sensors. This has been used for indoor navigation.

1.4 Literature review

Research is being carried out in two directions in the field of mobile robots. One is related to the mobile robots intended for outdoor applications and the other for indoor applications.

The following literature relates to the research in mobile vehicles used for outdoor applications. Ishikawa et al.[1] and Mcvey et al.[2] used a geometric model of the guideline to navigate the vehicle along a specific path. Herbert and Kanade[3] presented an active vision system using range data from a laser range scanner for outdoor scene analysis. Zheng et al.[5] introduced an approach to build a qualitative description of scenes along a route.

Research in indoor mobile robots originated with the work of Moravek[6],[7] for Stanford cart mobile robot. Next came Hilare designed by Giralt et al.[8] and Chatila and Laumond[9]. More recently the Mobi robot developed by Kriegman et al.[10] uses a pair of stereo cameras for 3-D vision. The stereo images are used both for building simplified maps of the environment and subsequent navigation through the environment. Another recent system by Ayache et. al.[11] uses extended Kalman filtering to help a robot acquire robust estimates of visual features in its environment. Kosaka and Kok[12] recently developed a vision based indoor mobile robot using model based reasoning and prediction of uncertainties.

There is voluminous amount of literature on the subject of path planning. The book by Latombe[13] a collection of relevant information needed. Suffice it to say that there now exist two major approaches for path planning, one based on configuration

space ideas pioneered by Lozano Perez and Wesley [14] and the other using potential functions, their used first proposed by Khatib[15] in the context of real time robot control.

1.5 Objective and Scope of the Present Work

An attempt has been made , in the present work, to build a model of a mobile robot, Mobile Platform, with the scope of carrying out the following activities; perception, planning and control, and actuation (as explained in the previous section), of the mobile robots. Figure 1.1 shows the the hardware setup of the present work.

A Moving Platform (Vehicle) has been built with two non-steerable independently driven wheels mounted on the same axis. This platform is of skid-steer type, Kanayama and Harman[16]. These two wheels are powered by two separate D.C servo motors. The platform has one passive wheel (castor) at the front end. The D.C motors are controlled by connecting them to a driver board which in turn is connected to the Microprocessor of a host P.C (PC1) (Intel 80386). The Driver board has the necessary electronic circuitry for providing the required power from a voltage source and to control the direction of the two motors independently by the Microprocessor. A program (Back ground control module) has been coded for controlling the two motors independently. The control of motors with only B.G (Back Ground) control module forms open loop control, since there is no feed back from the motors.

A typical path planning problem for a moving platform is , given initial position and orientation, and a goal position and orientation of the platform $\mathcal{A}$ in workspace $\mathcal{W}$, generate a path $\mathcal{T}$ specifying a continuous sequence of positions and orientations of $\mathcal{A}$ avoiding contact with the obstacles, $\mathcal{B}_i$'s, starting at the initial position and orientation, and terminating at the goal position and orientation.In the present work a program has been developed (Path planning module) which takes the information of workspace, robot, obstacles, initial position of the platform, and goal position of the platform needed as inputs and gives a path from initial position to goal position, without touching obstacles, in the form of connected lines. This program is used as an off-line path planner.

In the present problem the sensing (feedback) of the configuration, position and orientation, of platform has been established by Computer Vision. An Image Processing System (Imager-LC) using a C.C.D camera (30 Frames/sec) mounted on a

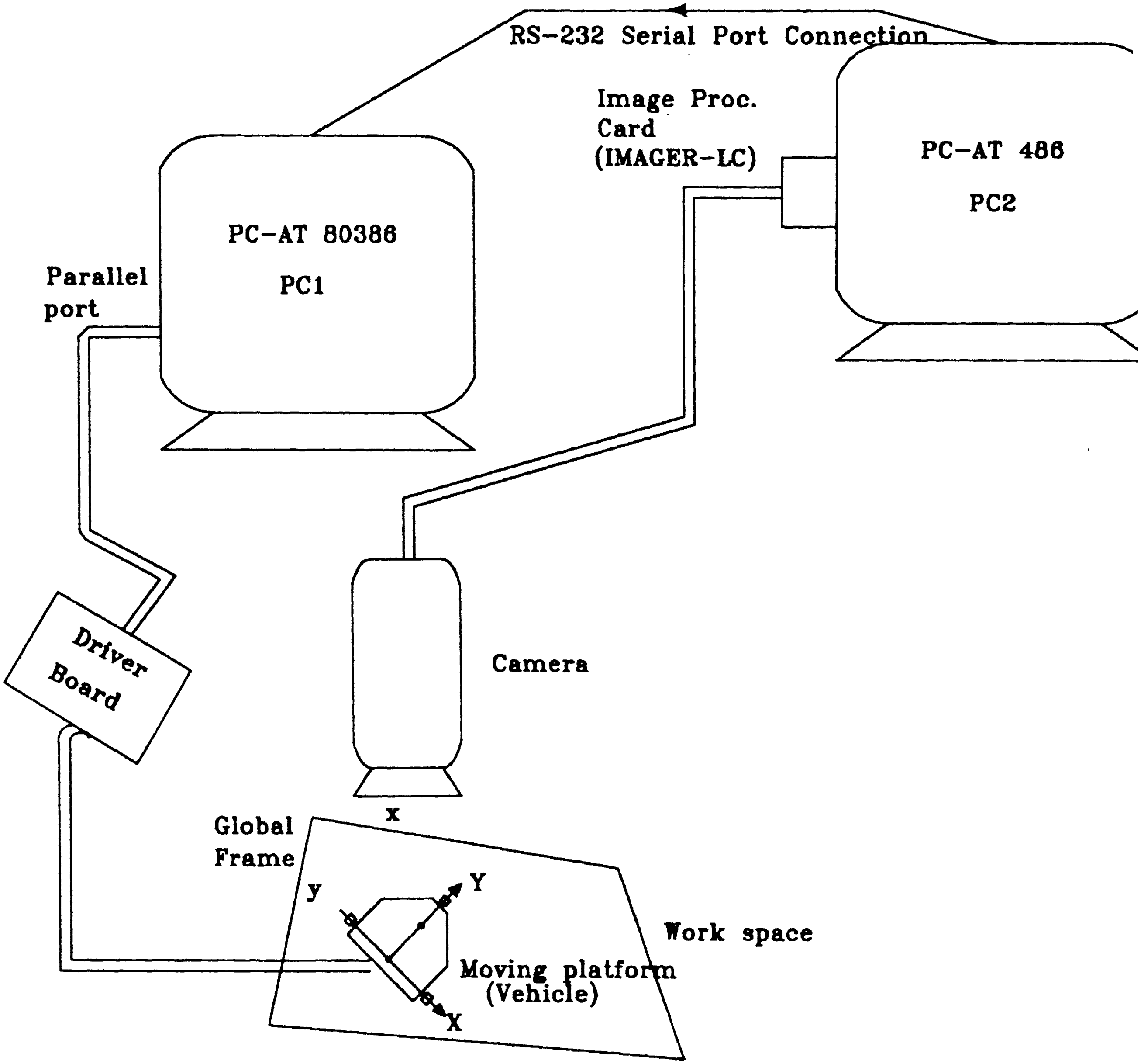


Figure 1.1: Hardware set-up

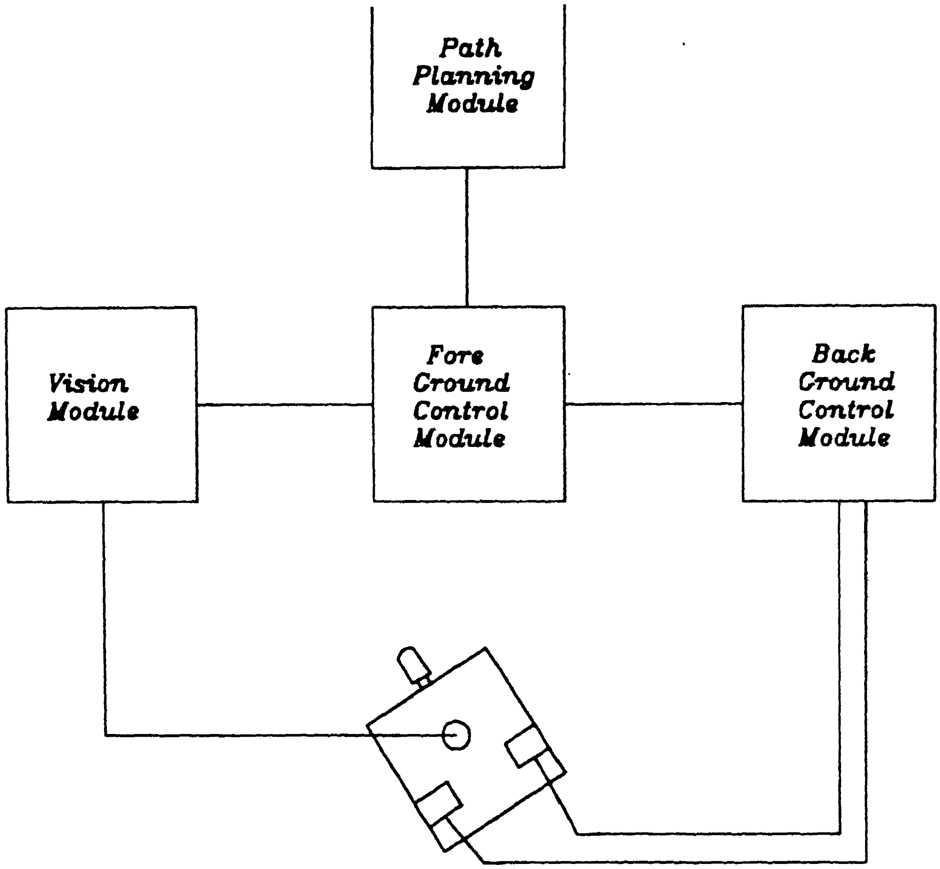


Figure 1.2: Software structure

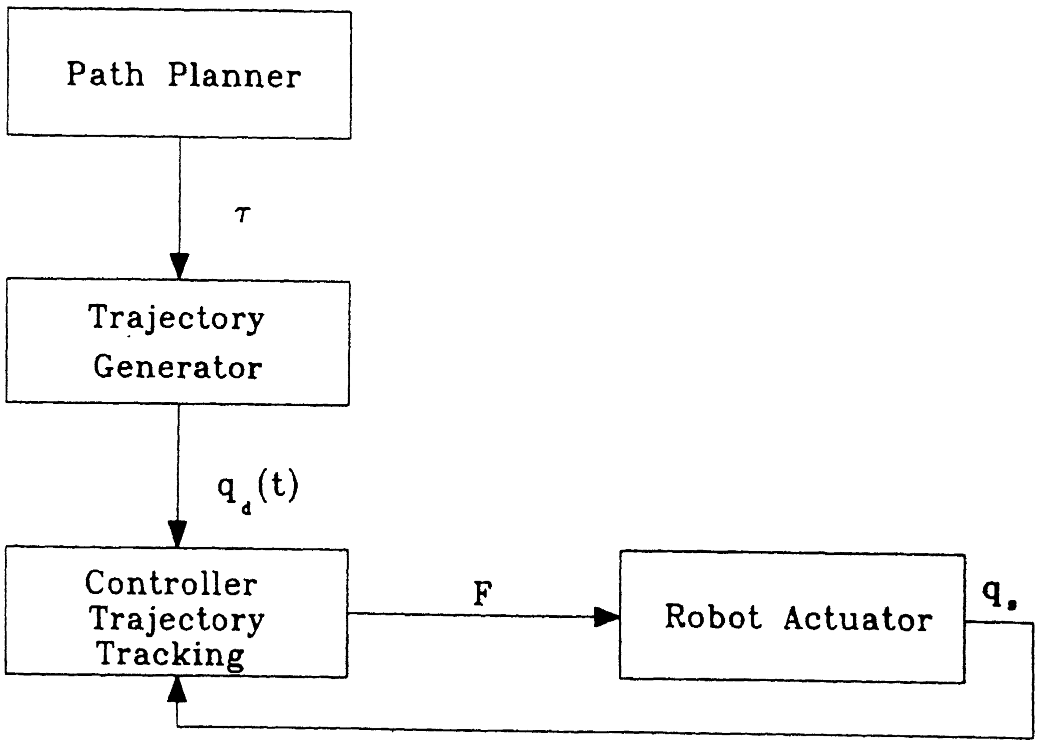


Figure 1.3: Closed loop control structure of a mobile robot

personnel computer, PC2 (Shown in fig.1.1), grabs an image and analyses it to find the configuration of platform in real time and transfer that information to the controller. A feature tracking algorithm (Vision Module) has been implemented, which grabs images of the workspace by the camera, detects the configuration of platform and transfers those configuration values to the controller (whenever it receives a request from controller) in a loop, for Dynamic path following. In a trajectory-tracking problem, the desired time history of the output variables is specified. Therefore in this case, the task is not only to reach a point, but also to reach it at a specified time instant. In Dynamic path following problem, however, the geometry of the path is specified. In this case it is more important to follow the path closely than to reach points on the path at specified time constants. Automatic recognition of the features on the platform ,at the beginning of execution, has not been implemented. RS-232 port was used for transferring the information to the controller, which is the P.C-386 to which the platform has been connected.

A controller program (Fore Ground control module) has been coded. Which is executed by the P.C1, to which platform has been connected shown in figure 1.1., in parallel with the B.G module.This program integrates the data from path planning module, vision module and controls the platform through Back Ground control module. The entire S/W structure has been shown in figure 1.2. In this way the requirements for the real time control has been established.

1.6 Organization of the Thesis

In Appendix A the Kinematic and Dynamic modeling and control of a Mobile Robot is presented briefly. Chapter 2 discusses the importance of visual motion and an algorithm for finding the configuration of the Moving Platform in real time. The control of the Moving Platform using P.C and detailed description of the driver board electronic circuit is presented in chapter 3. Also an algorithm for controlling the motors using P.C is presented.

Chapter 4 gives an introduction to Path planning and Path planning methods and a Path planning algorithm used in the present work. H/W and S/W details of the present work are mentioned in chapter 5. Chapter 5 also gives the details of a program which integrates the data of all the above mentioned programs together for implementing the entire work setup.

The H/W details of the Image Processing system are dealt in Appendix B. Appendix C briefly explains about the PWM control. The cable connection details of the entire work setup are presented in Appendix D, in the form of tables.

Chapter 2

Vision

2.1 Vision and The Computer

This chapter describes the importance of machine vision, an introduction to digital image systems, and various aspects of visual motion. An algorithm has been presented for tracking the moving platform in real time towards the end of the chapter.

Machine vision has opened exciting new worlds for man-machine interaction. The ability to see allows the machine to make decisions based upon the presentation of visual imagery, a task often taken for granted by humans. Further more a computer can act as a tool for the enhancement, restoration and storage of images for human interpretation. Indeed, the computers's capacity to see, represents a major step in the evolution of machine intelligence.

2.2 Digital Image Processing

Digital image processing, where the image information is processed digitally rather than by other means such as optics or analog circuits, found its first major use in the U.S. space program. Mainframe computers were used for image processing computations while special purpose electronics provided the storage and display of the digital image on a T.V. monitor. Commercial digital image processing systems began appearing in the market in mid seventies. These machines provided the necessary hardware for capturing images from standard T.V. signals as well as storing and displaying them. Software to carry out many common image processing functions provided digital image processing technology to the industrial user.

Vision systems must be capable of image sensing, image analysis and image inter-

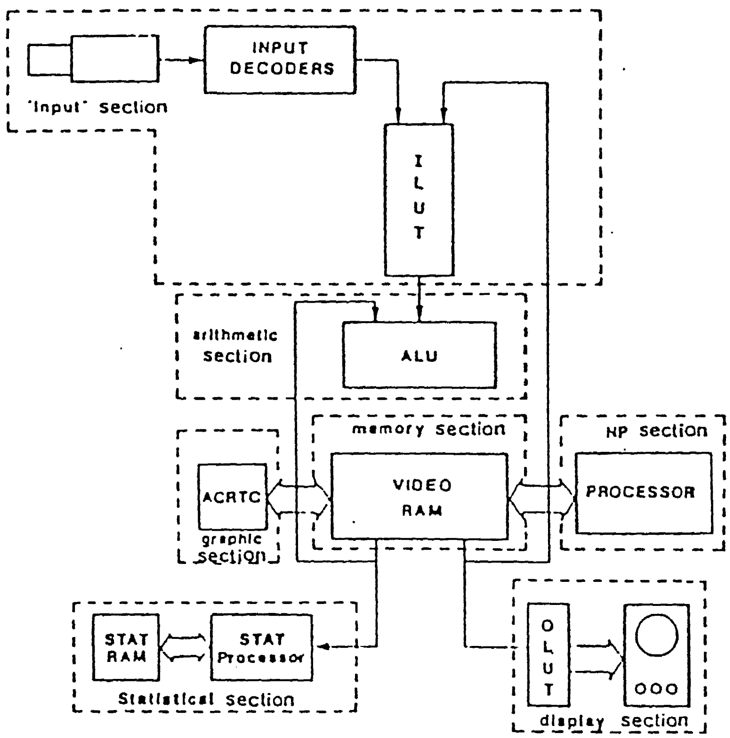


Figure 2.1: Block diagram of image processing system

pretation. Flexible image processing systems are capable of automatically acquiring information about an object, measuring its features, recognizing an object within a scene and making decisions based on the acquired data.

In an image processing system video cameras are linked to computers, picking up patterns of gray level intensities. The resulting image can be processed by a computer to perform enhancement or contrast operations for a higher quality image. The computer analyzes the images and extracts the required information such as the presence, position, orientation and identity of the object. H/W details of an image processing system are given in appendix B.

2.3 The Image Processing System

The structure of a typical Image Processing system is shown in figure 2.1. The image processing system, on which the present work has been carried out, is *Image-LC* . This Image-LC is a product of MATROX. The details of the Image-LC are briefly explained below.

1. Hardware : The Image-LC is a powerful single-slot frame grabber, image processing and graphics board set with on-board intelligence (10 MIPS 32-bit TI TMS34020 GSP) to keep the host CPU free for other tasks. The image-LC comes with an on-board asynchronous monochrome digitizer and real time processor as standard features. It hosts 4 Mbyte of local RAM and 3 Mbytes of configurable frame buffer memory (VRAM). The Image-LC has pseudo colour capabilities.

2. Software : The Image-LC runs using some essential programmable logic software and the Image series Image-Shell resident software. Once the firmware is downloaded on-board, one can use the Programmers Tool kit package to interface his C application programs with the Imager-LC. The Programmer's Tool kit is a layer of code that is executed on the host.

3. System Overview : The block diagram of the system is shown in Figure 2.1. A brief overview of each section is given below.

   a. Input section : The input section selects the video source from which it extracts video information. It is user-configured through software to match the devices connected to the front end of the Imager-LC.The section is composed of look up tables, and A/D converters, which complete the digitization function.

   b. Memory section : This section consists of 4 Mbytes of local RAM and 3 Mbytes configurable frame buffer memory. Its task is to hold image information.

   c. Statistical Section : This is a H/W section that performs intensity histograms and profiles.

   d. Graphics section : This section supplies many drawing primitives which allow the host to perform drawing routines with fewer commands.

   e. Display section : The responsibility of this section is to extract information from the memory, convert it to video and send it to the display.User defined display features available are Overlays, Windows, and keying.

   f. Arithmetic section : The arithmetic section performs all arithmetic and logical operations.

2.4 Motion Estimation

The study of visual motion is the study of how information about movement in an image can be used to make inferences about the structure and movement of the outside world. There are two parts to the problem. One is how the raw measurements of the changes produced by motion are made and the other is how this information can be used. The first part is called motion estimation. The second problem is to some extent a study of the minimum information necessary from the first part in order for subsequent computation delivers some sort of useful results. One of the most important problems in image sequence analysis is motion estimation. Some applications involving motion are computer vision, vehicle navigation and for compressing the data in T.V. signal transmission.

The existing methods of motion estimation can be divided into two categories namely Feature based methods and Pixel based methods. In both the methods motion is estimated in two steps. First 2-D motion is analyzed (feature based methods) or estimated (pixel based methods) and then 3-D motion is estimated,see table 2.1.

2.4.1 Feature Based Methods

In feature based methods, 2-D image features such as points and lines are selected and extracted from the image. Then the correspondence of the extracted features between the consecutive frames is found out.That is which item in the image at time t_1 corresponds to which item at time t_2 (t_1 and t_2 are timings at which two successive images of the work space have been taken), is found out. Finally using constraints such as rigid body motion, a system of equations in terms of the 3-D motion parameters (translation and rotation angles) is developed. The number of motion parameters must be equal to or less than the number of equations so that the system is solvable. This constraint will determine the minimum number of features that must be used. The vertices have been used in [19] and sharp changes in curvature in [20]. Features based on global properties have been used in [21] and [22].The Line correspondence has been used in [23, 24] and curve correspondence in [25].

The establishment and maintenance of feature correspondence is a very difficult problem. Occlusion may cause features to appear or disappear. Noise also creates difficulties in the extraction and correspondence of features.

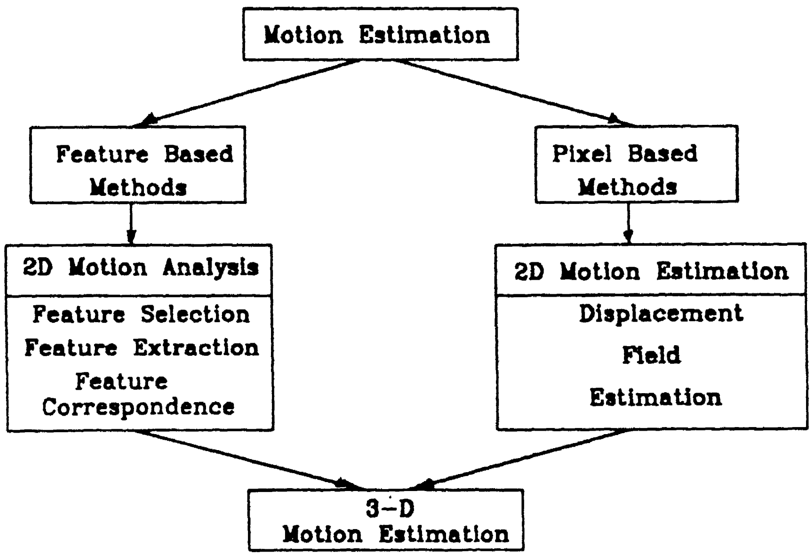


Table 2.1: Motion estimation methods

2.4.2 Pixel Based Methods

In pixel based methods the displacement of each pixel is estimated separately. The estimate is based on the relationship between time and spatial differences. For each pixel the displacement vector is estimated. This field of vectors is called the displacement field or optical flow.

The pixel based methods have some important drawbacks. They are very sensitive to noise since they need calculation of derivatives. They assume small change in displacement, which need a high rate for grabbing image frames. They also assume the motion field is smooth, an assumption that is wrong around boundaries. These methods are also computation intensive and thus need dedicated H/W for finding the optical flow.

2.5 Dynamic vision

Human beings, when talking about dynamic scenes, do not converse in image terms but prefer spatial interpretations both in position and velocity. They try to see motion of objects in space. Motion properties of an object are an integral part of humans knowledge base like possible shapes and colours. This means that not just objects are being seen but motion processes of objects in space and time. Note that unlike static image sequence processing, dynamic vision has no separation between spatial object recognition from one frame to next as a first step and motion reconstruction afterwards as a second one. Instead, object and motion are treated as a unit and some fit is used for determining the best estimate for object motion state, based on the noise corrupted image sequence is done in space and time simultaneously. As a side effect the need for storing past images (e.g for computation of displacement vector fields) is reduced.

An important concept upon which to base the design of a dynamic vision system is temporal continuity. Usually, natural scenes change gradually. If two images of such a scene are taken within a few milliseconds they will normally be very similar to each other. Assume that a first T.V. image of such a scene has just been interpreted. It is then rather easy to interpret the immediately following image as the differences between the two are very small. This observation has important consequences for the design of real-time vision systems. Therefore, the cycle time of the low level vision sub system should ideally be less than one frame period of the T.V. signal used, making it possible to evaluate every single image as it is delivered by the camera.

The appropriate behavior of a vision controlled machine typically depends on the presence and location, or absence of certain objects in its environment. The vision task is then clearly goal directed, the first sub task being to locate features in the image which are indicative of the presence of important objects. In many situations such features occupy only a small fraction of the total image area.It suffices then to process only those areas of each image which actually contain relevant features.

In dynamic scene interpretation the location of all important features is usually known in advance from the interpretation of previous images. This means that when interpreting the next image in sequence, the search space in which the features of interest should be looked for is small, and features can be rediscovered rather quickly. The most important point here is, since nearly all relevant information in the image is contained in a limited number of small regions the combined size of this region is only a small fraction (less than 10%) of the whole image. So all the available computing power can be concentrated on these regions. Different algorithms in each region, containing different type of features can be used, refer to Figure 2.2.

The concept of processing only a limited number of well defined regions within an image is also the key to a natural division of the problem into sub tasks which can be executed in parallel on a coarsely grained multi processor system.

2.5.1 Feature Detection and Tracking Algorithms

In dynamic vision the time for processing each image should ideally be less than one frame period. Two powerful approaches are available to maximize the speed of feature extraction. One is the application of advanced knowledge and the other is to strictly concentrate on obtaining only that information which is necessary to accomplish the given task.

These two approaches emphasize the difference between static image processing and dynamic vision. In static image processing very little is known in advance of the image presented to system. The task is then to extract as much information as possible. But in dynamic vision each new image is known to be a natural continuation of a sequence of images which the system has interpreted already. So the differences relative to the previous images are is expected to be in small details only, like how much and in which direction a feature moved in a small amount of time.

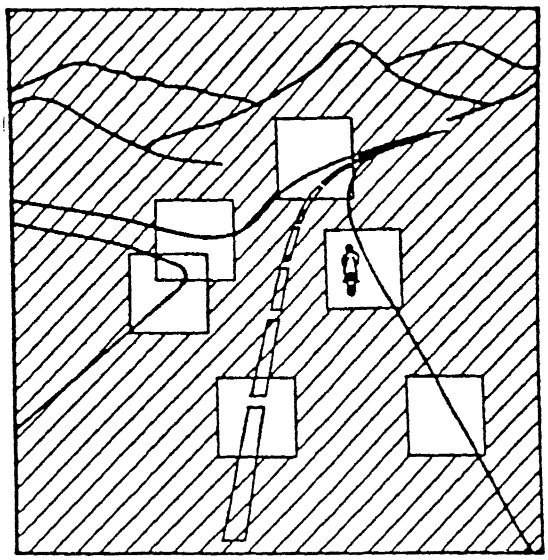


Figure 2.2: An image of an autonomous vehicle in a natural environment

2.5.2 Task Specific Feature Extraction

All the available resources in a dynamic vision system should be concentrated on obtaining that information which is necessary or at least helpful. For the present problem of controlling a mobile vehicle, since the vehicle is having 3-dof, three configuration variables , two for position and one for orientation (x, y and θ), are sufficient to represent the vehicle on the workspace. Coordinates of any two points on the vehicle as a function of time, with respect to a global frame, are sufficient to find the three variable (x, y and θ) at any time. If the coordinates of more points are available this provides valuable redundancy which can be used to make the system more robust. All the available computing power should, therefore, be concentrated on locating various points of the vehicle.

Referring to Figure 2.2, defining all features relevant for an autonomous vehicle in a natural environment is more difficult because of the great variety of situations it may encounter. Certain parts of the image can not contain any relevant information such as mountains and the sky as shown in Figure 2.2. Certain others will always be relevant, for instance, gray level edges which are characteristic of borders of the road or lane, other vehicles etc.

The key point here is that the low-level part of a dynamic vision system should process only those features which yield information usually required by the higher levels. The requirements of the higher levels are derived from desired performance of the machine to be controlled.

2.5.3 Knowledge Based Feature Extraction

As most features in typical scenes move by at most one or two pixels between two successive images, a *Zeroth order prediction* where it is assumed that the feature will reappear at the same location as in the last image will often be sufficient. In case of very fast moving objects a *First order prediction* which also takes the estimated velocity into account, may sometimes be more appropriate. For the present work First order prediction has been taken into account.

2.6 Feature tracking algorithm

Experimental setup of the present problem consists of c.c.d camera attached at the ceiling and the Vehicle moving on the ground, in the field view of the camera. Camera has been connected to an image processing card with a PC-AT 80486(PC2). Figure 1.1 shows the hardware setup.

In order to determine the configuration of the vehicle at any time (x, y and θ of a representative point) which is moving on the 2-D plane (ground is assumed to be flat, $z = 0$) only coordinates of two points on the vehicle, with respect to a global frame,as a function of time, are needed. To obtain coordinates of two points on the vehicle as a function of time, two circular marks or features have been created on the top of the vehicle, so that by tracking those two features on-line and by finding centers of those two features, the coordinates of two points on the vehicle as a function of time can be found out.

2.6.1 Algorithm

1. Create a search window which is contains the circular feature, marked on the top of the vehicle, with its sides slightly bigger than the diameter of the circular feature. This search window is called region of interest (ROI). Refer to fig. 2.3(a).

2. Find the histogram of the ROI.Refer to fig.2.3(b).
3. Decide the threshold value to make the pixels corresponding to feature completely black (gray level value zero) and that of background completely white (gray level value 255).
4. Select an LUT (Look Up Table) and fill it with the values shown as a function in fig.2.3(c).
5. Map the ROI with selected LUT to get black and white image with black region belongs to feature and white region to background.
6. Erode and dilate the ROI to remove any noisy pixels.
7. Find the profile of ROI both in X and Y direction, shown in fig. 2.3(d).
8. Search the arrays containing X and Y profiles for finding the square region on the image plane containing the circular feature and having the length of the side equal to the diameter of the circular feature.
9. This newly acquired center will form the center point of the search window (ROI) for the next cycle.
10. Go back to first step.

The same algorithm has been implemented for the other feature also, so that the two points, needed for finding the configuration (Position and Orientation) of the Moving Platform with respect to the Reference Frame, are known.

A special feature of this algorithm is that it contains only simple operations (additions, subtractions etc..) and hardware implemented statistical image processing functions. Thus the cycle time is small, so that the two features on the vehicle can be tracked in real time.

2.7 Serial Port Transmission

RS-232 serial port transmission has been used for transferring the coordinate values of the two points on the vehicle on-line from PC2 to PC1. There are basically two commonly used serial protocols in the laboratory with IBM-PC and compatible, RS-232-C and RS-232-D. RS-232 standard,shown in fig.2.4, was designed to allow computers to be interfaced to terminals via commercial telephone lines. The device used to translate signals between the telephone lines and serial line coming from computer is called

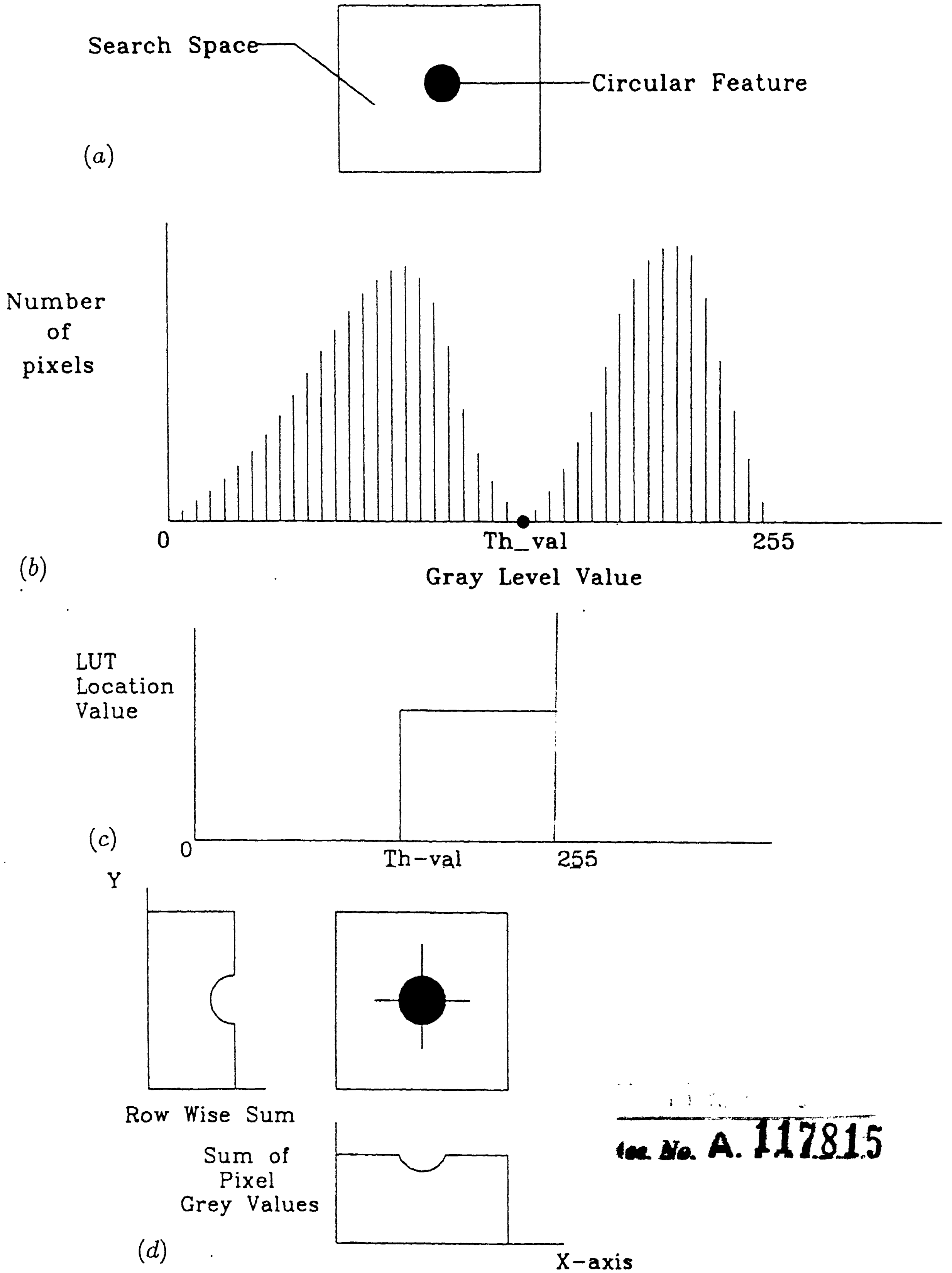


Figure 2.3: (a) A circular feature contained in a search space (b) Histogram of the search space (c) Mapping function for making the search space of image into B/W (d) X and Y direction profiles of the

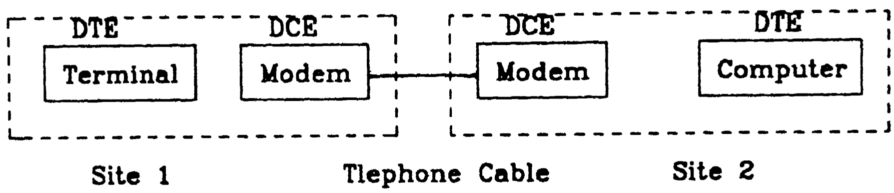


Figure 2.4: Typical remote terminal system

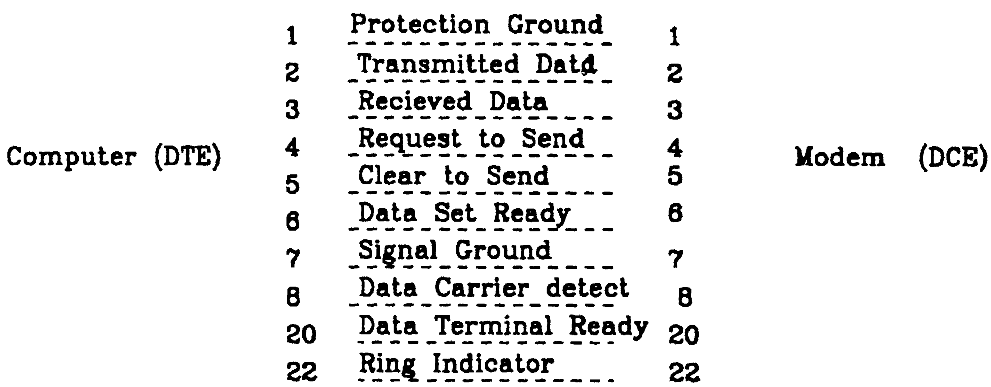


Fig.1.6 RS–232–c Connections

Figure 2.5: RS-232 connection details

Modem. Refer to figure 2.5 for RS-232-C connections. But RS-232 interface has been adopted by a number of manufacturers to a variety of non modem applications namely connection between a micro-computer and a plotter or printer, between two computers (as shown in Figure 2.6). These are technically called null modem applications. Communication between the two P.C's can be achieved by cross connecting the transmission and receive wires of the two P.C's. The details of the transmission are:

Data transmission rate(Baud rate)	9600 bits/sec.
parity	even
No. of data bits	8
No. of stop bits	1

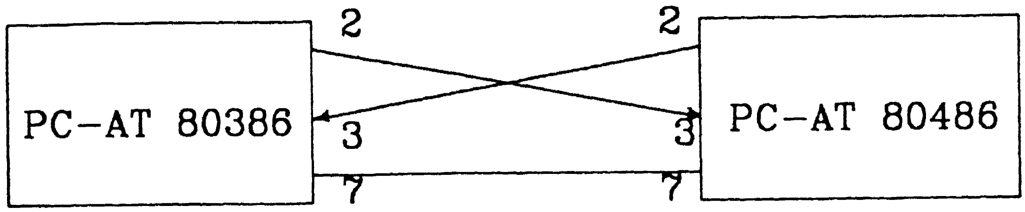


Figure 2.6: Serial port communication between two computers

2.7.1 Algorithm for Serial Port Transmission

Serial port transmission has been used for transferring information from PC2 to PC1. This serial port communication forms a link between vision module and foreground control module.

1. Set baud rate of the serial port to 9600.
2. Set even parity check .
3. Set number of parity bits to one.
4. Find the center coordinates of two features created on the vehicle as discussed in the previous algorithm.
5. Check for a request from PC1.
6. If there is any request waiting from PC-AT 80386, execute from step 7 to else continue from step 4.
7. Wait until transmit register becomes vacant.
8. Transmit lower byte of the X coordinate of the center of first feature.
9. Wait until transmit register becomes vacant.
10. Transmit higher byte of the X coordinate of the center of first feature.
11. Continue the steps from 7 to 10, for transferring Y coordinate of the center of first feature and X and Y coordinates of the center of second feature.
12. Continue from step 4.

Chapter 3

Control

3.1 Introduction

This chapter focuses on the control part of the present work. Both direction and velocity control of the motors of the vehicle are discussed in detail. An algorithm for controlling the motors has been presented at the end.

3.2 Selection of Motors

There are various power drives available which could be used for locomotion of mobile vehicles. Some of them are Gasoline engines, pneumatic and hydraulic drives, and electric motors. Of these electric motors are preferred because of their better speed regulation, control by microprocessors, and maintenance free operation. In any case, the selection of a power drive mainly depends on the specific task in hand. For the present problem gear head D.C. micro motors have been used.

3.2.1 D.C. Motor

In case of D.C. motors torque is created according to Laplace law, whereby a conductor placed in a magnetic field is subjected to a force proportional to the current. A D.C motor is characterized by the following relationships.

1. Torque is proportional to current.
2. Speed is a function of voltage.

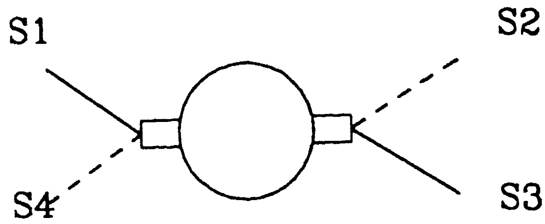


Figure 3.1: Simple direction control of a motor

It is a traction motor: as long as the power is supplied it provides torque and there is no holding torque. It acts as an active motor in servo systems. To place a D.C. motor under servo control, a speed sensor and position sensor must be used. The details of the motors used for the present problem can be found in chapter 5.

3.3 Electronic Controller

3.3.1 Direction Control of a Motor

The drive circuit implements the configuration given in Figure 3.1. It consist of electronic switches SW1,SW2,SW3,SW4; realized by transisters. The complete circuit diagram is given in Figure 3. 2). There are two sets of identical switches $SW1/SW3$ and $SW2/SW4$. The former set of switches provide clockwise rotation where as the later set provide counter clockwise rotation. The following explanation is with respect to motor $\bar{A}$, as shown in Figure 3.3 , . The control signals for BAF and BAR are supplied by pulses (logic levels $0V = LOW$, $5V = HIGH$) through the printer port of a PC-AT 80386 by software control. The input of the NAND gate is passed on to the output when the protection bit ,P, is high, with the output being the inverted input. Let us consider the motor A rotating in the clockwise direction, for this to happen the following logic levels are needed.

$$P = HIGH, BAF = HIGH, BAR = LOW.$$

Then

$$N1_{o/p} = N4_{o/p} = LOW N2_{o/p} = N3_{o/p} = HIGH$$

The high NAND outputs make the concerned transistors ON, where as the others remain OFF. Thus for the above logic levels transistors T1 and T4 are OFF, and T2

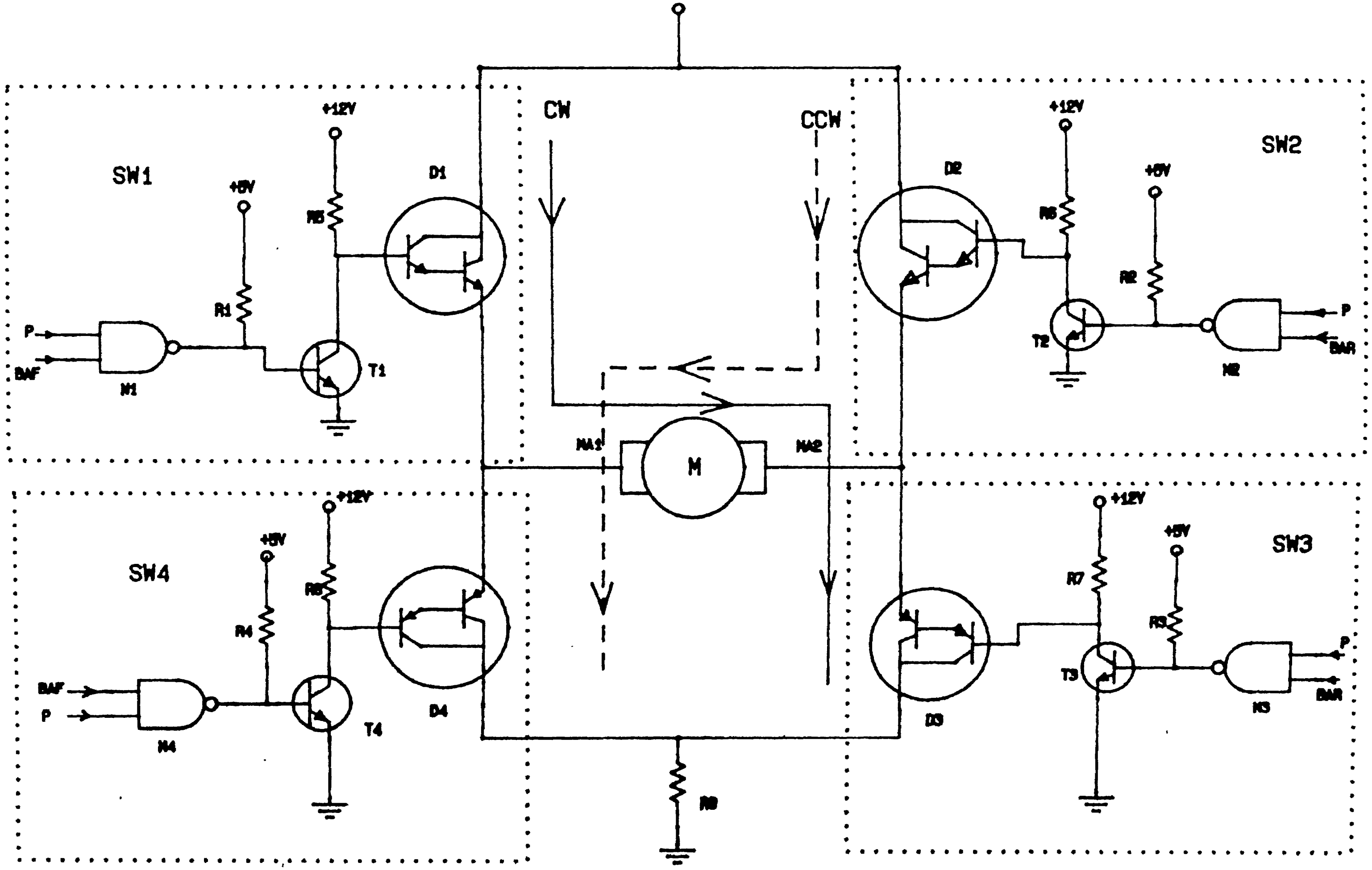


Figure 3.2: Driver board

Electronic components list

B	74LS244 OCTAL BUFFER LINE DRIV
N	74LS38 OPEN COLLECTOR NAND
T1,T2,T3,T4	SL100 NPN TRANSISTORS
D1,D2	BD677 NPN DARLINGTON PAIR
D3,D4	BD678 PNP DARLINGTON PAIR
R1,R2,R3,R4	4.7k (1/4 W)
R5,R6,R7,R8	1k (1/4 W)

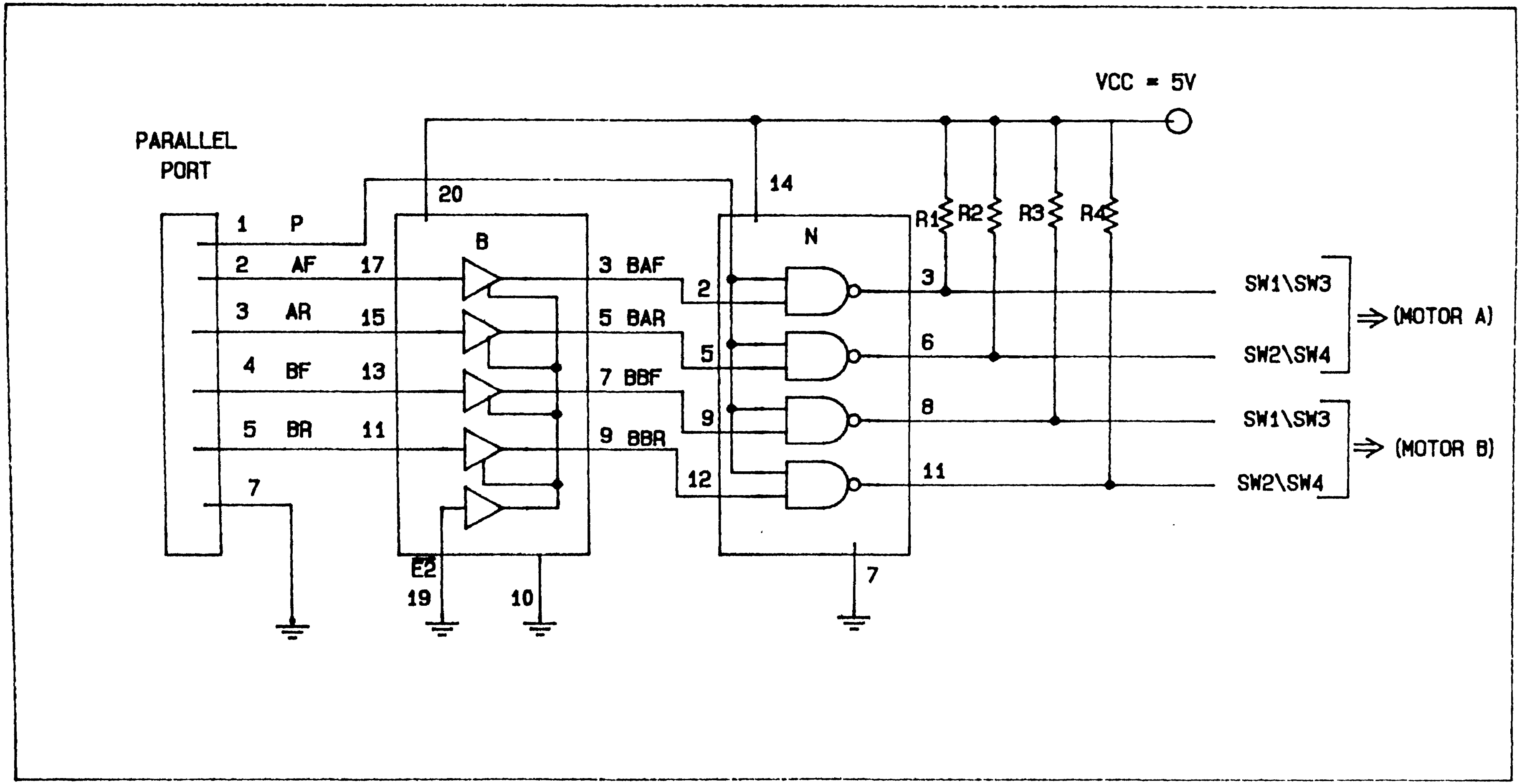


Figure 3.3: Printer port connection to driver port

Table 3.1: Table showing the states of the transistors for forward and reverse rotations of the motor

Rotation	Enable	BAF	BAR	N1o/p	T_1	D_1	N2o/p	T_2
Forward	H	H	L	L	OFF	ON	H	ON
Reverse	H	L	H	H	ON	OFF	L	OFF

Rotation	D_2	N3o/p	T3	$D3$	$N4o/p$	T_4	D_4	M A_1	M A_2
Forward	OFF	H	ON	ON	L	OFF	OFF	24V	GND
Reverse	ON	L	OFF	OFF	H	ON	ON	GND	24V

and T3 are ON. As a result the collector current flows through the transistors T2 and T3, which causes their collector potential go low.The opposite is the case with other two transistors T1 and T4,that is their collector potential is at 24V. In the Figure 3.3(a), D1,D2,D3, and D4 are darlington pairs. D1 and D2 are NPN type and D3 and D4 are PNP type. For a PNP transistor the base voltage should be at lower potential than the emitter potential to make it ON where as for an NPN transistor it should be at higher potential. In a darlington pair the current of the first transistor causes h_{fe} (forward current gain of a transistor) times base current through the emitter which forms the base current for the second transistor. This base current is further amplified by the second transistor by h_{fe} times and as a result a heavy current amplification is obtained. The ON/OFF states of the darlington pairs depend on the corresponding states of the transistors T1,T2,T3 and T4. The transistor ON/OFF states for forward and reverse rotation of the motors are given in the Table 3.1.

The duty cycle (Will be explained latter) of the pulses decide the speed of the motors. By using the above mentioned electronic controller both direction and velocity of a motor can be controlled. Two such controllers are needed for controlling two motors independently.

Following the above discussion, AF, AR and BF, BR are the two sets of control bits for two motors, to decide the direction of current flow. Logic Table 3.2 shows the corresponding motor status.

Table 3.2: Logic table

Forward	Reverse	Motor Status
0	0	NOP
0	1	CCW
1	0	CW
1	1	NOP

$$\begin{aligned} NOP &= Nooperation \\ CW &= ClockWise \\ CCW &= CounterClockWise \end{aligned}$$

3.3.2 Velocity Control of a Motor

It is often useful to control the voltage across a motor in such a way that only two values are used, usually zero and one. The usefulness derives from two sources:

1. The easiest signal to produce with a computer is a binary signal and
2. The most efficient way to modulate power flow is with amplifiers that can only operate at FULL-ON or FULL-OFF. One way to achieve the appearance of continuous speed control in this concern is Pulse-Width-Modulation (explained in APPENDIX B) technique.

A pulse-width-modulated signal is a constant frequency, two valued signal in which the proportion of the period for which the signal is ON and the proportion of the period for which the signal is OFF can be varied, as shown in Figure 3.4. The percentage of ON time is called DUTY CYCLE.The details of PWM are given in APPENDIX C.

For pulse width modulation to be effective, the chosen frequency of the PWM must be high enough so that the motor should not have time to respond to the rapid ON/OFF changes, but instead, the speed of the motor should reflect the average power level over many cycles. This average power level is controlled by the duty cycle of the PWM signal. Thus, PWM can be effectly used to control the speed of motors.

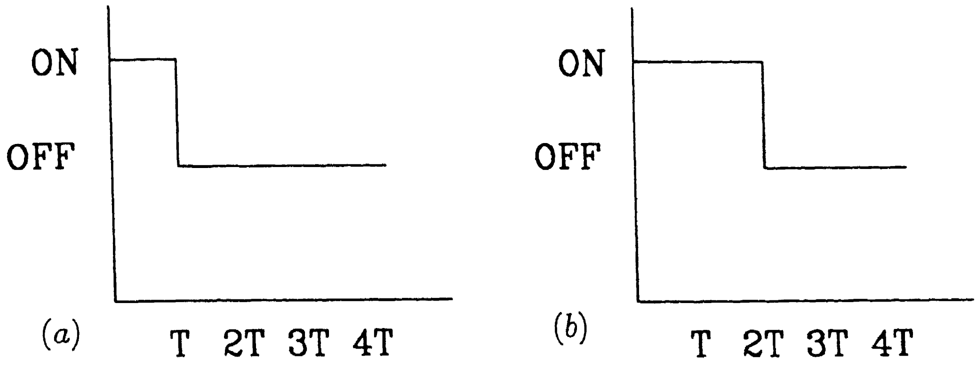


Figure 3.4: (a) 25% duty cycle. (b) 50% duty cycle

PWM using Timer Interrupts

The PC1 computer uses vectored interrupts. Whenever a device interrupts the processor, it is responsible for telling the CPU which interrupt service routine should be executed. In other words the external device provides a vector that points to its own service routine. An interrupt vector table is maintained by the CPU; there are 256 entries in the table, and each vector entry contains the address of an interrupt service routine, as shown in the Table 3.3.

If a program wishes to catch interrupts, it must first choose an interrupt vector, note the address of the existing interrupt service routine residing in the corresponding entry of the vector table and put the address of an interrupt service routine in that entry. The original interrupt service routine (already existing address) should be chained when ever the interrupt comes. On PC-AT 80386 the external device vectors range from 8 to 15, and the timer uses vector number 8.

TURBO C provide two functions, getvect() and setvect(), to read existing address of an interrupt service routine from the vector table and to write an address into the table. Getvect function takes one argument, vector number of the vector table, and returns the already residing interrupt service routine address. Setvect function takes two arguments, one is the entry number of vector table and the other is interrupt service routine address. This setvect function simply put the given interrupt service address at the given entry of the vector table. While executing, these two functions prevents the CPU from responding to externally generated interrupts when a new vector entry is being installed. Also these two functions restore the original contents of the vector entry table whenever the program exits.

All external device interrupt requests are first processed by two 8259A, programmable interrupt controllers, cascaded together.This 8259A interrupt controller is designed to provide a powerful but simple interface between external devices and CPU. Each 8259A can handle 8 interrupt request inputs, and each request is associated with a priority and a vector table number. These are set by the operating system.

The PC-AT 80386 system has three programmable timers controlled by an INTEL 8254-2 timer / counter chip and defined as channel 0 through channel 2. Channel 0 is the system timer. The input of the system timer is connected to 1.19 M.Hz OSC where the output frequency is programmable by dividing the input frequency with a suitable number. The output of 8254-2 is connected to IRQ-0 of 8259-A. The output frequency of 8259-A is 18 ticks/sec. So the CPU gets an interrupt by timer every 55.07 m.secs.

When an external device requests an interrupt by signaling one of the 8259's interrupt request input lines, the CPU looks at all interrupt requests and selects the one with highest priority. If there is no interrupt routine currently being serviced or if the priority of the request is higher than that of the routine currently being serviced, the 8259-A will pass the interrupt request onto the CPU.

On receipt of an interrupt request, the CPU stops execution after the current instruction and enters the *Interrupt Acknowledge Sequence* . During this phase the 8259-A sends the vector number to the CPU and marks the request as currently being serviced.The CPU saves the current state of the flag register and its current place in the code segment. Then, the CPU uses vector number to index into the interrupt vector table and obtain the address of interrupt service routine. This interrupt service routine is then executed. The completion of the current service routine is signaled by sending an END-OF-INTERRUPT (EOI) control byte to 8259-A. The CPU then retrieves the memory address where it was interrupted and continues the execution.

The observation here is that, since the execution time for interrupt service routine is very small, the two programs being executed , one in foreground and the other in background, operate in near real time. The foreground program is the one which has been interrupted by the external device and the background program is the one supplied by external device. In the present case the actual generation of the PWM signal is accomplished by the background program. Any changes in DUTY CYCLE of PWM signal can then be affected through foreground program.

Table 3.3: Interrupt vector table

Vector Number	Interrupt Service Routine Number
1	x x x x x 1
2	x x x x x 2
3	x x x x x 3
4	x x x x x 4
5	x x x x x 5
6	x x x x x 6
7	x x x x x 7
8	x x x x x 8
256	x x x x x 256

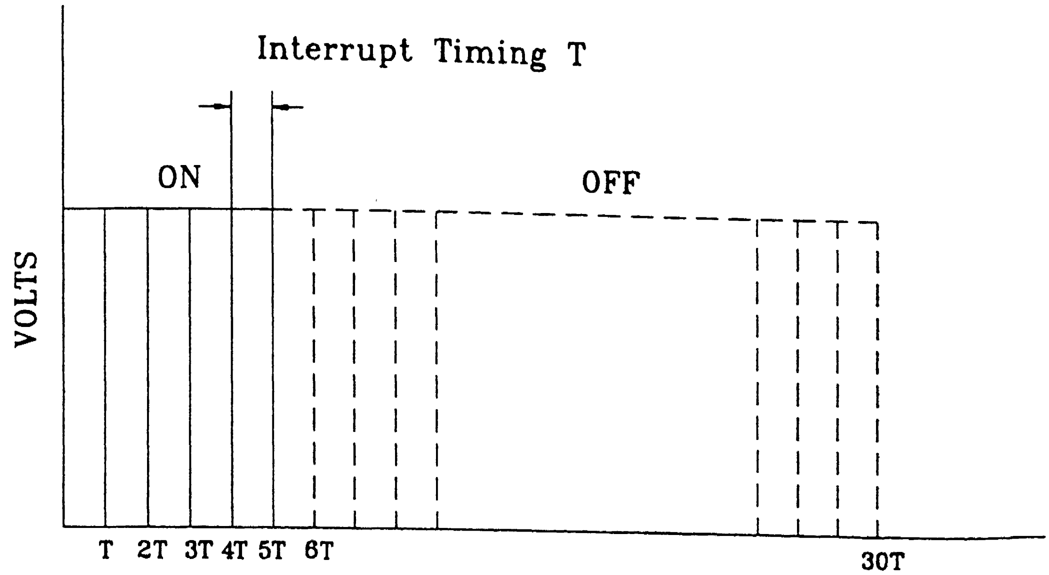


Figure 3.5: Interrupt control

Timing Analysis

By dividing the system timer input frequency by 266, an output frequency of 2.236 K.Hz can be obtained. Since this output frequency is connected to 8259A, it will be interrupted 2236 times per second. So the CPU executes the interrupt service routine, background program, 2236 times per second. The Time period (T) of PWM should be fixed at the beginning of the program. In the present problem the Time period (T) is fixed to the time taken by 30 consecutive interrupts.

Let the time taken from the beginning of one interrupt to the beginning of next interrupt be t sec. Then Time period T = 30 * t sec.

The duty cycle of PWM is manipulated through foreground program while CPU is not executing background program. This duty cycle will be fixed at the beginning of each cycle (after every 30 consecutive interrupts). So any changes made in duty cycle will be operational from the next cycle. Each time the background program is executed, it checks whether the PWM should be ON or OFF based on the duty cycle and accordingly sends ON or OFF signals to motors. Suppose the duty cycle is 20%, then PWM signal should be ON for a time of 5t sec and OFF for a time of 25t sec, for a total Time period of 30t sec as shown in the Figure 3.5.

3.4 Algorithm for B.G Control Module

This module will be executed when ever the CPU gets interrupted by the system timer. The time period of PWM signal has been fixed to the time taken by 30 consecutive timer interrupts. An integer variable PULSES was declared and initialized to zero for repeating the PWM cycle after every 30 interrupts. Two integer variables ON and OFF were declared for controlling the duty cycle of PWM.

The direction control of the two motors has been achieved using four bits. A bit can take two values either zero or one. The value one represents rotation state of a motor (either clockwise or anti-clockwise) and the value zero represents the idle state of a motor. So four bits are needed for controlling two motors both in reverse and forward direction (one bit for each direction). The direction control of the two motors was realized through the printer port of PC1. The first four bits of the printer port character were chosen for this purpose. Logic table is shown in 3.1.

1. If $PULSES < ON$, do step 2 to step 6 Else do step 7 to 11.
2. Get the directions needed for the first and second motors from F.G control module.
3. Perform logical AND operation on relevant bits of the printer port character, CUR_CMD, for both motors with the values obtained from step 2.
4. Send the character CUR_CMD to printer port.
5. Increment the value of PULSES by one.
6. Execute previous interrupt routine (timer service routine) and return to F.G control module.
7. Perform logical AND operation on relevant bits of the printer port character, CUR_CMD, for both motors with zero's.
8. Send the character to printer port.
9. Increment the value of PULSES by one.
10. If the value of $PULSES > 30$, equate PULSES to zero.
11. Execute the previous interrupt routine (timer service routine) and return to F.G control module.

Chapter 4

Path Planning

4.1 Introduction

Creating autonomous vehicles is a major undertaking in Robotics. It definitely requires that the ability to plan motions automatically be developed. Indeed, except in limited and carefully engineered environments, it is not realistic to anticipate and explicitly describe to the autonomous vehicle all the possible motions that it may have to execute in order to accomplish required tasks. Automatic path planning allows the user to specify tasks more declaratively by stating what one wants done rather than how to do it.

4.2 Configuration Space

The underlying idea of configuration space is to represent the vehicle as a point in an appropriate space, the vehicle 's configuration space, and to map the obstacles in this space. This mapping transforms the problem of planning the motion of a dimensioned object into the problem of planning the motion of a point.

Let the robot $\mathcal{A}$ (at a certain position and orientation) be described as a compact (i.e closed and bounded) subset of $\mathcal{W} = R^{\mathcal{N}}, \mathcal{N} = 2$, and obstacles $\mathcal{B}_1, \ldots, \mathcal{B}_q$ be closed subsets of $\mathcal{W}$. Let $\mathcal{F}_a$ and $\mathcal{F}_b$ be cartesian frames embedded in $\mathcal{A}$ and $\mathcal{W}$,as shown in fig.4.1, respectively. $\mathcal{F}_a$ is a moving frame while $\mathcal{F}_w$ is fixed one. Since $\mathcal{A}$ is rigid, every point a of $\mathcal{A}$ has a fixed position with respect to $\mathcal{F}_a$. But $\mathcal{A}'$s position in $\mathcal{W}$ depends on the position and orientation of $\mathcal{F}_a$ relative to $\mathcal{F}_w$.Since $\mathcal{B}_i$, for all i includes $\{1, q\}$, are both rigid and fixed in $\mathcal{W}$, they have a fixed position with respect to $\mathcal{F}_w$.

A configuration of an arbitrary object is a specification of the position of every

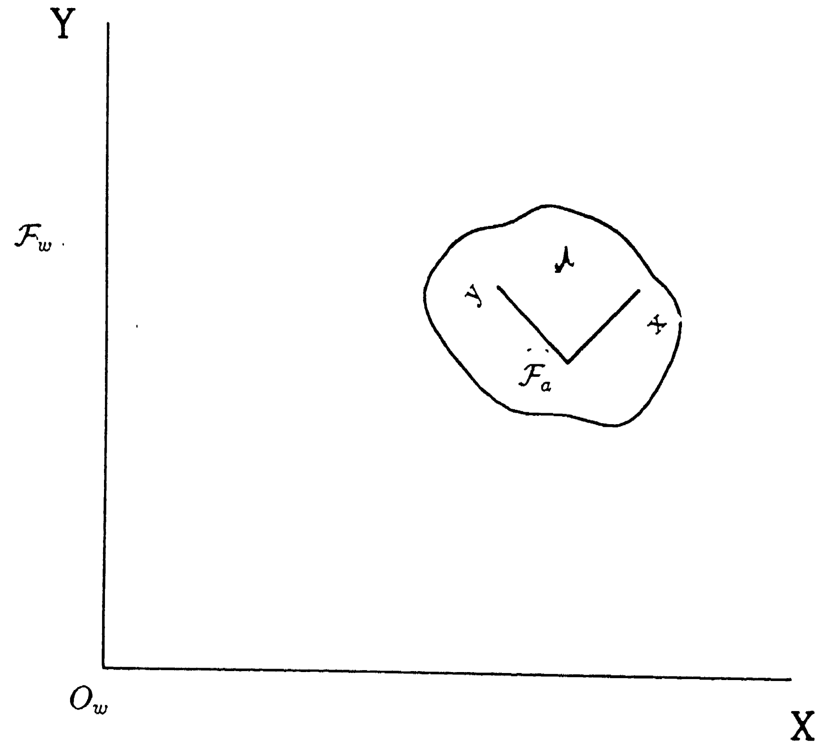


Figure 4.1: Frame convention for Moving Platform

point on this object relative to a fixed reference frame.Therefore, a configuration of robot $\mathcal{A}$ is a specification of the position and orientation of $\mathcal{F}_a$ with respect to $\mathcal{F}_w$. The configuration space $\mathcal{C}$ of an object is all the configurations of the object. Every obstacle $\mathcal{B}_i$, i = 1 to q, in the work space $\mathcal{W}$ maps in $\mathcal{C}$ to a region:
$\mathcal{CB}_i = \{q \in C \setminus \mathcal{A}(q) \cap \mathcal{B}_i \neq 0\}$
which is called a C-obstacle. The union of all C-obstacles:
$\bigcup_{i=1}^{q} \mathcal{CB}_i$
is called the C-OBSTACLE region.and the set:
$\mathcal{C}_{free} = \mathcal{C} \setminus \bigcup_{i=1}^{q} \mathcal{CB}_i$
is called the free space.

4.3 Path planning approaches

There exist a large number of methods for solving the basic motion planning problem. Despite many external differences, the methods are based on a few general approaches. Some of these are Road maps, Decomposition methods, and Potential field methods.

Currently Decomposition methods and Potential field methods are increasingly used.

4.3.1 Decomposition Methods

These methods consist of decomposing the free space (intersection of workspace and c-space obstacles) into simple well behaved regions called cells as shown in figure 4.4, such that a path between any two configurations can be easily generated. A non-directed graph, representing the adjacency relation between the cells is then constructed and searched. This graph is called the connectivity graph. Its nodes are cells extracted from the free space and two cells are connected by a link if and only if the corresponding cells are adjacent. The outcome of the search is a sequence of cells called a channel. A continuous path can then be computed from this sequence.

4.3.2 Potential fields

In this method the vehicle, represented as a point in configuration space, is a particle moving under the influence of an artificial potential produced by the goal configuration and c-space obstacles, as shown in figure 4.3. Typically the goal configuration generates an attraction potential, which pulls the vehicle towards the goal and c-space obstacles produce a repulsive potential, which pushes the vehicle away from them. The negative gradient of the total potential is treated as an artificial force applied to the vehicle at any instant. At every instant the direction of this force is considered the most promising direction of motion.

Unlike decomposition methods, potential field methods do not include an initial processing step aimed at capturing the connectivity of the free space in a concise representation. Potential field methods are called local methods and decomposition methods are called global methods. The major draw back of potential field methods is they can get trapped into a local minima of the potential function different from the goal configuration.

Decomposition methods have been chosen for the present problem because it simplifies control. The path obtained by decomposition methods consist of a set of connected straight lines as shown in figure 4.4.

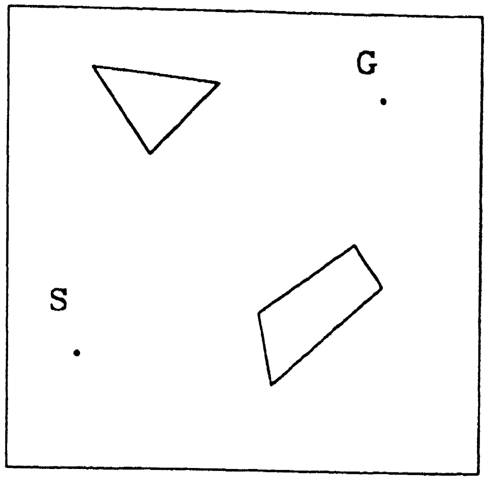


Figure 4.2: Path planning problem

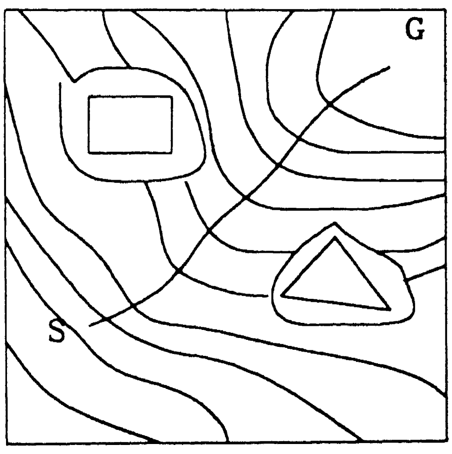


Figure 4.3: Potential field method for path planning

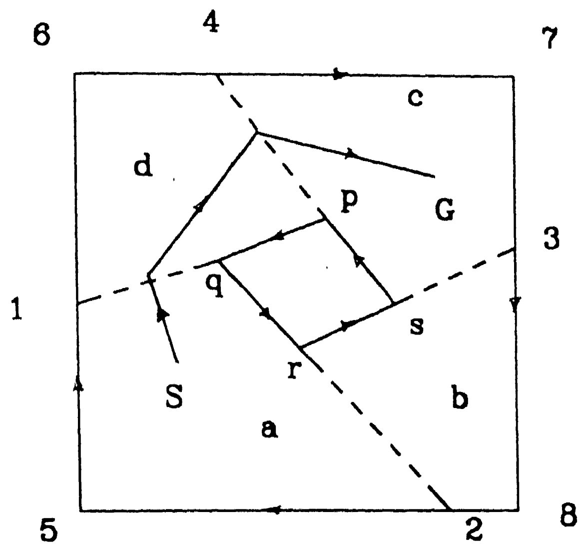


Figure 4.4: Decomposition method for path planning

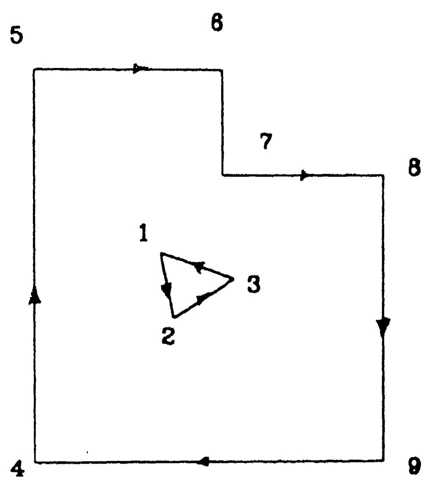


Figure 4.5: Work space containing a triangular obstacle and a moving platform

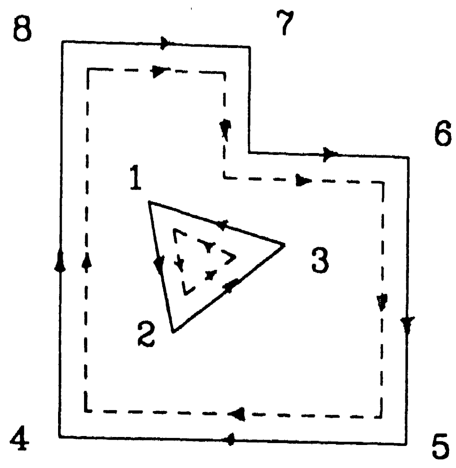


Figure 4.6: C-space mapping

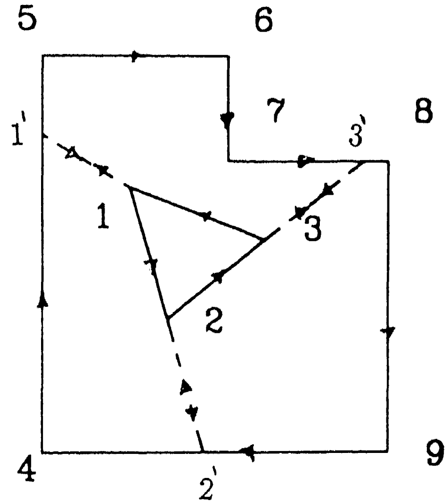


Figure 4.7: Decomposing the work space into convex regions

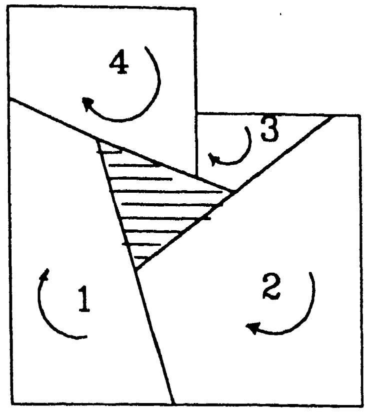


Figure 4.8: Decomposed convex regions

4.4 Convex decomposition method

In this method the vehicle,c-space obstacles, and wall are approximated as polygons. The vehicle free space is divided into convex regions or convex cells. The basic advantage of convex decomposition is that, within a region from one point to any other point vehicle can move in a straight path.so the path obtained consists of minimum number of turnings, which is highly advantageous for the present problem.

4.4.1 Algorithm for convex decomposition

1. Given coordinates of the Vehicle,Obstacles and wall in workspace figure 4.5, map obstacles and wall in the configuration space C of the vehicle to obtain the C-obstacles, and the C-wall.
2. Label all edges of the wall and obstacles such that free space is always to the right figure 4.6.
3. For all edges, starting at any edge do
4. If the turn to the next edge, in anti clockwise direction, is less than PIE radians, then extend the edge until it intersects some other edge as shown in figure 4.7.
5. Reorder the edge list:
   old edge becomes longer,

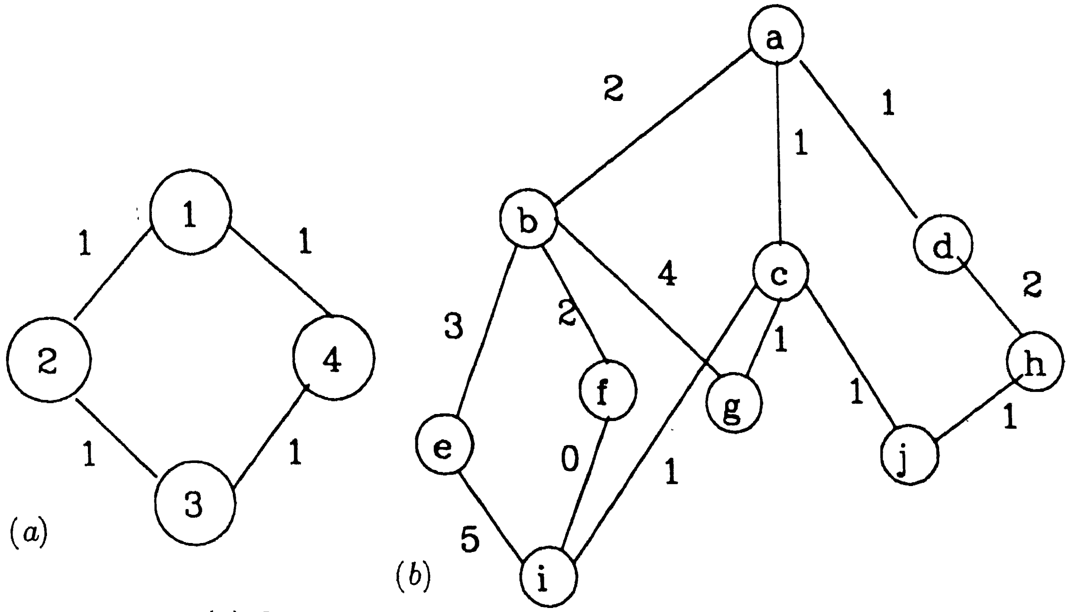


Figure 4.9: (a) Graph formed by the decomposed regions (b) Graph search

add extension edge in reverse direction,
split up the distal edge into two edges.

6. continue from step 3, until all edges have been traversed.
7. Form closed convex regions in the free space, by searching the above modified list of edges as shown in figure 4.8.

The above algorithm decomposes the free space into convex regions.

4.4.2 Algorithm for Graph Search (DIJKSTRA)

Let B = (Define) Start node. G = (Define) Goal node. FRING = Set of nodes about to be expanded. TREE = Set of nodes already expanded.

1. Form a connectivity graph with each convex region as a node as shown in figure 4.9.
2. Get start point from the user and decide to which region it belongs.
3. Take goal point from user.
4. FRING = $\{B\}$
5. TREE = $\{\}$
6. If G includes TREE, goto step 9
   else if FRING == $\{\}$ exit FAIL.
7. For all vertices belong to FRING, find vertex V such that
   Weight (PARENT, V) + dist (PARENT) is minimum.

FRING = fringemerge($\{FRING - V\}$,$\{children(V)\}$)
TREE = union (TREE,V).

8. Continue from step 6.

9. Find a straight line path from start point to goal point, from the channel of graph nodes found above and exit success.

Algorithms for the Functions used above:

- Weight(PARENT,V)
  return(The cost or weight of link between PARENT and V)

- dist(PARENT)
  return(sum of all weights from start node to PARENT)

- fringmerge(s1,s2)
  If a member c belongs to s2 is also in s1, adjust PARENT of c according to shortest distance otherwise add c to s1.

- union(TREE,V)
  Adds member V to TREE list.

For the present problem equal weightage has been given for all links. This provides a solution which will have minimum number of connected straight lines.

Chapter 5

Implementation and Results

5.1 Hardware Details

The entire H/W set up can be divided into three groups namely Vision related H/W, Control related H/W and Vehicle H/W.

5.1.1 Vision related H/W

1. A C.C.D camera,Figure 5.1, along with a 220V / 12V transformer, camera control box and a zoom lens.
2. Imager-LC, image processing card as shown in Figure 5.2.
3. A PC-AT 80486(PC2) Computer for housing the Image processing card.
4. Flat cable to connect serial ports of PC2 and PC1.
5. Nine pin D type connectors, two numbers, for serial port connection.

5.1.2 Control related H/W

1. Driver Board, shown in Figures 3.2 and 5.3.
2. A PC-AT 80386 (PC1) Computer.
3. A flat cable to connect printer port and Direction controller of motors.
4. Regulated 24V D.C supply 5.5.
5. A flat cable to connect Driver Board and vehicle.
6. Twenty five pin D type connectors (male), two numbers.

7. Twenty five pin D type connectors (female), two numbers.
8. Nine pin D type connectors, two numbers.
9. A fifty pin F.R.C. connector.

5.1.3 Vehicle H/W

1. Gear head D.C motors shown in Figure 5.4, two numbers.
2. Left and right wheels of the vehicle of diameter 5.5 cm. and width 4 m.m, made of aluminum shown in Figure 5.6(b).
3. Setup for fixing a twenty five pin D type connector on the body of the vehicle.
4. A plate shown in Figure 5.6(a), which forms the body of vehicle for mounting motors and other accessory.
5. A Caster wheel Figure 5.6(c), which forms the front wheel of vehicle.
6. A rectangular plate of 28 c.m * 32 c.m, for marking two circular features on the top of the vehicle.
7. Mounting for keeping the camera on the top of the vehicle for autonomous navigation (This problem has not been solved in present work).

5.1.4 Driver Board

Driver board is an electronic circuit having necessary elements for amplifying the input low voltage signal such that it can drive 24 volt D.C. motors. It has the necessary arrangements for moving the motors both in forward and reverse directions.

5.2 Software Details

All the code has been developed using Borland C compiler and linking with Matrox Imaging Library. The entire Software has been grouped into four categories as shown in figure 1.2.

1. Vision module.
2. Fore Ground Control module (F.G. Control module).
3. Back Ground Control module (B.G. Control module).

Figure 5.1: C.C.D. camera

Figure 5.2: Imager-LC Image processing card

Figure 5.3: Driver board

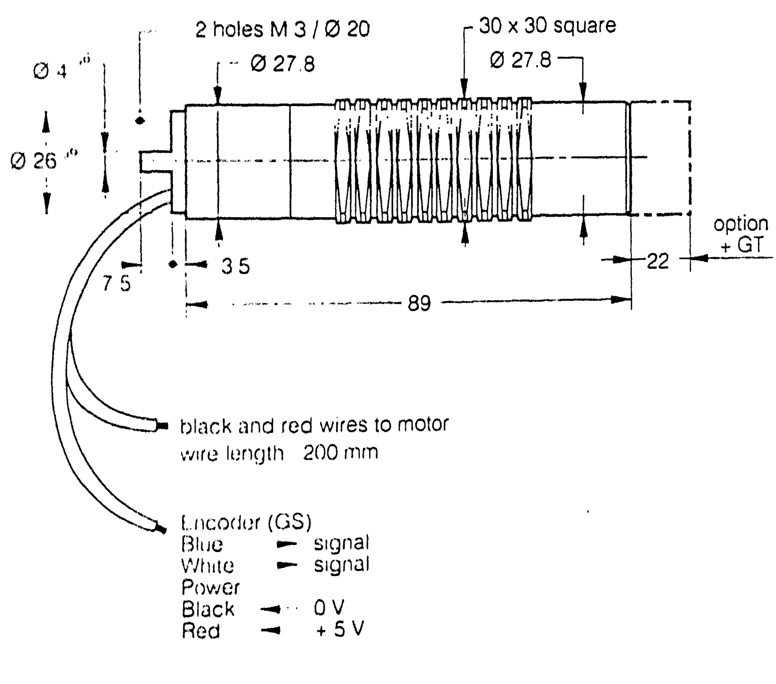


Figure 5.4: A D.C. motor

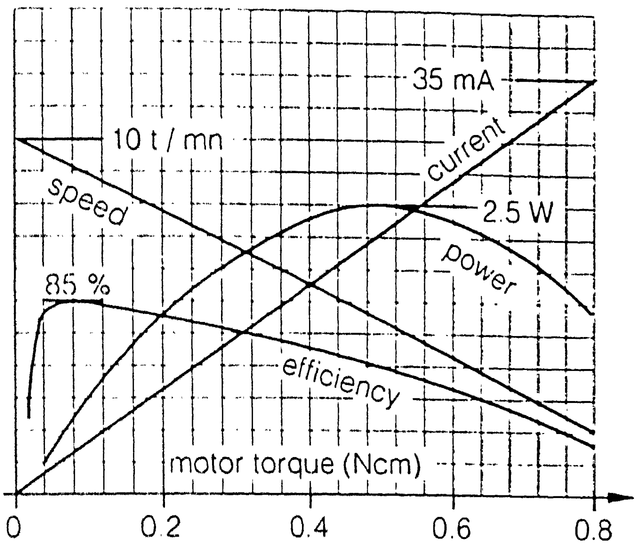


Figure 5.5: Motor characteristic curves

4. Path planning module.

First three categories work ON-LINE, where as the fourth one works OFF-LINE. vision module is executed on PC2 and all other modules on PC1.

1. Vision Module: This module is executed on PC2 . An Image processing card (Imager-LC) has been installed on the PC2, as mentioned in chapter 2. The algorithm for this module is discussed in chapter 2. This module takes the image of the work space as input and process the image to get the position and orientation of the vehicle in real time. These values are then supplied to the F.G. control module by RS-232 serial port.

2. Path Planning module: This module takes the work space coordinates, obstacle coordinates, start and goal point coordinates as input from the user. It finds the path to be followed by the vehicle without touching the obstacles. This path is then supplied to the F.G. control module.

3. Fore Ground control module (F.G. control module): This module acts as a S/W controller as shown in figure 5.8. The path to be followed by the vehicle is supplied to this module (in the form of connected lines) at the

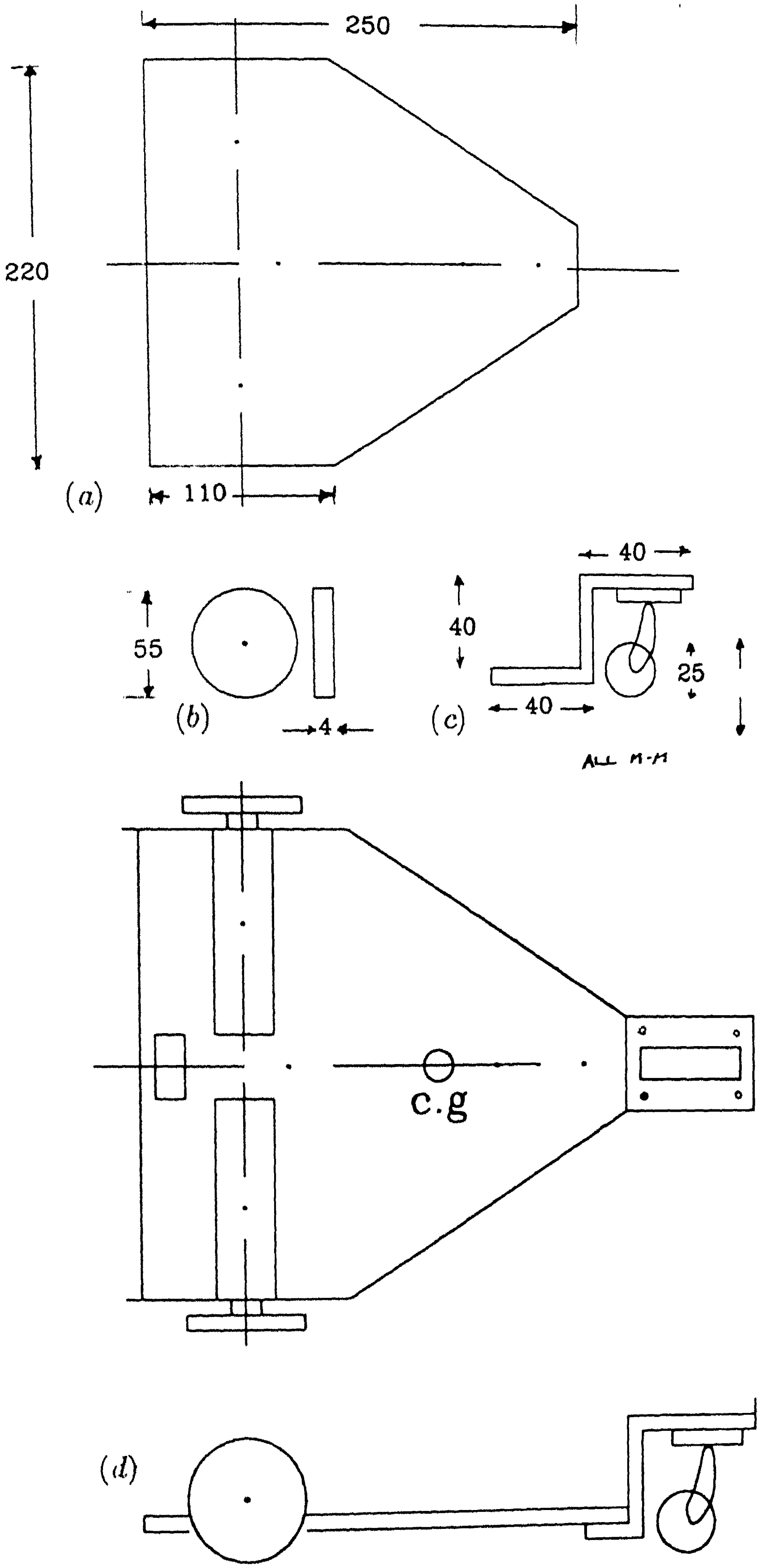


Figure 5.6: (a) Body plate (b) Wheel (c) Caster (passive wheel) (d) Vehicle assembly

Figure 5.7: Hardware set-up

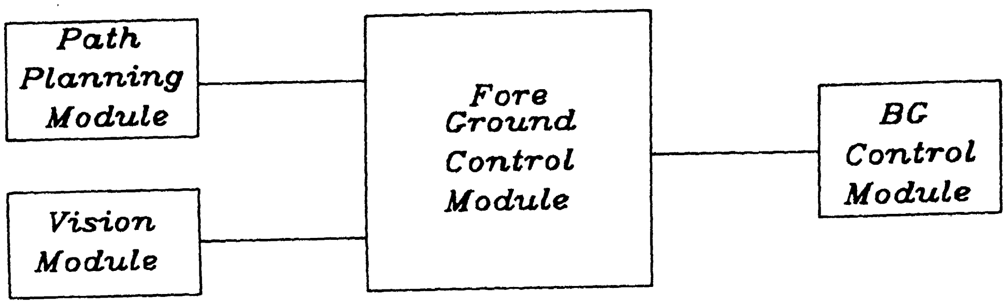


Figure 5.8: Fore ground control module

beginning of the execution. The above mentioned input forms the Reference input. The position and orientation of vehicle is supplied to this module as feed back input from the Vision module. Then this module finds the error between the path to be followed by the vehicle and current position of the vehicle. Depending on the error a decision will be taken on how much voltage should be supplied to each motor, such that the vehicle will follow the given path. Those values of voltages and directions are then supplied to the B.G. control module, for controlling the motors.

4. Back Ground control module (B.G. control module) This module takes the control parameters calculated in F.G. control module and sends it to the motors. It gets the modified values of voltage and direction of motors from F.G. control module in each cycle. Those changes are then sent to the motors through printer port.

5.3 F.G. control module

The functions of this module have been discussed in the previous section. The input from path planning module consists of a set of connected straight lines from start point to goal point, OFF-LINE. The other input from vision module consists of the values of configuration variables (x, y and θ) ON-LINE. So, the problem of guiding the vehicle from start point to goal point simplifies to guiding the vehicle along few straight lines. If the vehicle can be guided along a single straight line from one end to the other end, the same procedure can be implemented for all other lines until the goal point is reached.

5.4 Straight Line Following

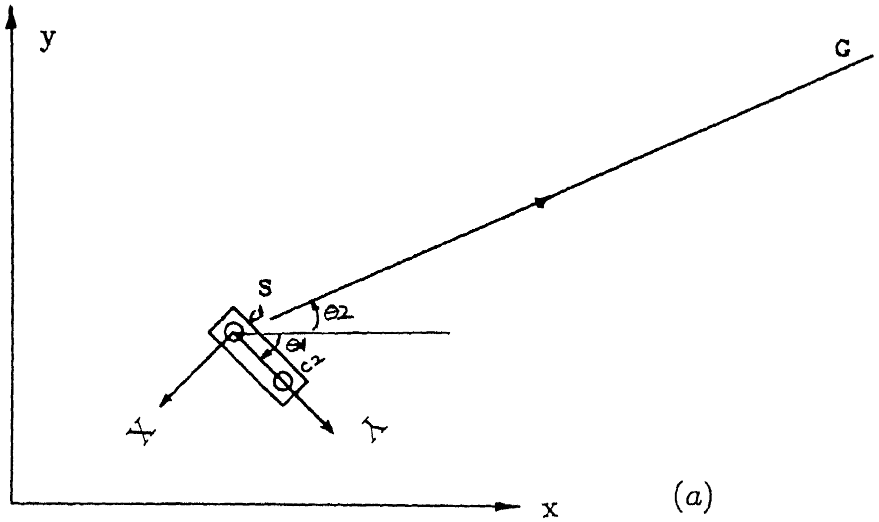
For the vehicle to follow a straight line two functions are needed. The first function guides the vehicle to match its direction with that of the line to be followed. The second function guides the vehicle to follow the given straight line such that the shortest distance from the vehicle's mid point (Origin of the robot centered frame) to the line being tracked is always minimum.

The first function takes the direction of the vehicle (θ, with respect to global x - axis) as ON-LINE input. It also takes the direction of the line to be followed (angle of the line with respect to the global x - axis) as OFF-LINE input.Then the error angle , difference in the above two inputs (both in magnitude and direction) is calculated.The magnitude of voltage and direction of rotation needed for the two motors are decided depending on the error angle as shown in Figure 5.9(a) and they are supplied to the B.G. control module for realizing them in the motors. The above mentioned ON-LINE actions continue until the vehicle direction matches with that of the line to be followed.

The second function takes the shortest distance, P, as shown in Figure 5.9(b), of the origin of the robot centered frame to the line to be followed as ON-LINE input. Vehicle needs a right turn while moving in forward direction, if the value of 'P' is positive. It needs a left turn while moving in forward direction if the value of 'P' is negative. Base voltage is the voltage supplied to the motors which provides minimum velocity to the vehicle while moving in forward direction. Turning voltage is the voltage supplied to one of the motors in addition to Base voltage, such that the vehicle takes right or left turn while moving in forward direction. The Base and Turning voltages are then calculated depending on the value of 'P'and supplied to the B.G. control module for controlling the motors.

5.5 Results

A moving Platform has been built with the components mentioned in section 5.1 Driver board and other related cables are made with the connection details as shown in Appendix D. Programs, as mentioned in section 5.2, were coded and executed successfully. The entire set up is used for implementing the problem of straight line following as described in section 5.4 and then extended to cover the path planning module discussed in section 4.4. This straight line following is basically a dynamic path following problem,



(a)

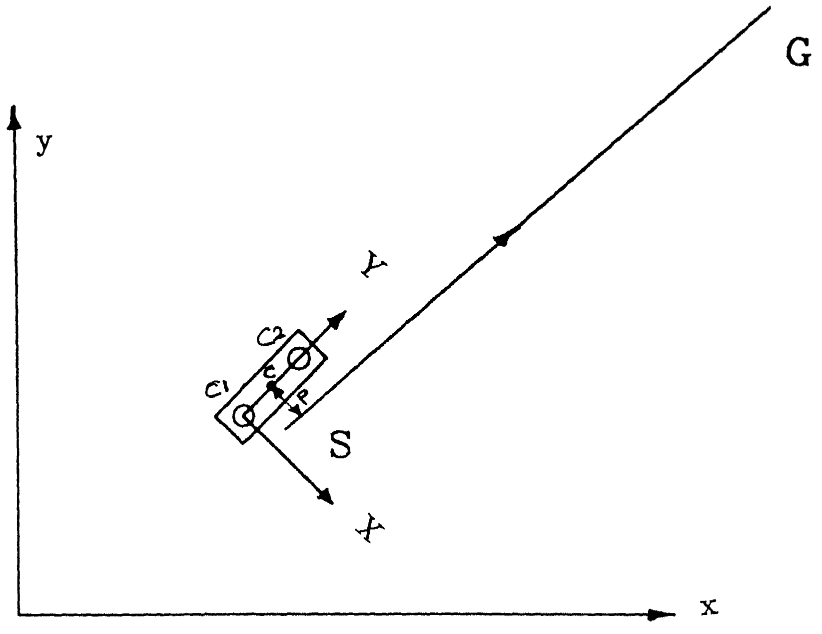
S	Start point
G	Goal point
C1	Center point of the first feature on the Platform
C2	Center point of the second feature on the Platform

E	Error angle
V1	Voltage supplied to Motor-1
V2	Voltage supplied to Motor-2
$\theta 1$	Direction of Platform with respect to x axis.
$\theta 2$	Direction of line to be followed w.r.t x axis.

$$E = (\theta 1 - \theta 2)$$

$$V1 = -V2 = K(E)$$

$$\begin{aligned} K(E) &= 4 \; (E > 10) \\ &= 2 \; (3 \leq E \leq 10) \\ &= 0 \; (-3 < E < 3) \\ &= -2 \; (-10 \leq E \leq -3) \\ &= -4 \; (E < -10) \end{aligned}$$



(b)

C	Center point of the two circular features of the Platform
P	Perpendecular distance of C from the line to be tracked
(V1 - V2)	= g(P)
Minimum(V1,V2)	= Base voltage
g(P)	$= 6$ ($P > 10$) $= 3$ ($2 \leq P \leq 10$) $= 0$ ($-2 < P < 2$) $= -3$ ($-10 \leq P \leq -2$) $= -6$ ($P < -10$)

Figure 5.9: (a) Matching direction of the platform to that of a straight line (b) Straight line tracking

since it does not involve any time constraint. The accuracy with which the Platform follows close to the straight line depended on the base voltage supplied to the motors. Two results are shown with different base voltages supplied to the motors in figure 5.10.

Chapter 6

Conclusions and Suggested Extensions

The capabilities needed for mobile robots as discussed in the introduction have been to some extent incorporated into the Moving Platform in the present work. Building a fulfledged mobile robot needs many sensors for acquiring and manipulating a substantial model of its operating environment. In the present work only a C.C.D.Camera (vision sensor) has been used for obtaining feed back of configuration variables. This issue of world modeling has not been covered in the present work and it can be extended to real world modeling by adding other necessary sensors and integrating all of them together.

This resulting world model can serve as a basis for crucial operations such as path planning, dynamic obstacle avoidance, landmark identification and navigation. These operations are very helpful in doing projects based on autonomous navigation.

A typical mission for a mobile robot can be described by a sequence of via points at which the robot comes to rest in a given configuration (position and orientation) to perform a given task (manipulation, sensing etc.). A simplified version of the above problem has been solved in the present work. It consists of given start and goal points, and work space with obstacles, a path consisting of connected straight lines is found and the platform is guided along that path. So while following any straight line the platform will have only one direction. Also time optimal motions have not been considered. This work can be extended to the more complex problem of finding time optimal motions of the platform in cartesian space and corresponding control trajectories that will move the robot from an initial configuration to a final configuration avoiding obstacles. Literature related to the above work can be found in

the work of David B.Reister [26] and Nilanjan Sarkar [27].

Necessary hardware, for mounting the camera on top of the platform and using it for outdoor applications, has been made with a view of extending the present work. Software has to be developed for using the above mentioned set up. Related information can be found in Ishikawa[1] and [4].

Appendix A

A.1 Mobile Robot Modeling and Control

The development of kinematic and dynamic models for analyzing and designing robot manipulators was one of the earliest contributions in robotics. These models are essential for robot trajectory planning and for controlling manipulator motions.

Although mobile robot research dates back to the late 1960's and early 1970's , only recently have similar modeling issues been addressed in the context of mobile platforms. Muir and Neuman [17] report on kinematic modeling of wheeled mobile robots and in a second paper, discussed dynamic modeling of robotic mechanisms. Careful modeling of vehicle dynamics and vehicular interaction with the road are also an integral part of the road following work discussed by Dickmann [18]. The following are some of the differences between kinematic modeling of wheeled robots and manipulators.

1. A wheeled mobile robot always has more than one wheel in contact with the surface it is travelling over and, consequently, is modeled as a multiple closed link chain.

2. The wheel of a mobile robot can both turn and translate with respect to the contact point between it and the floor. This pseudo joint is described as a higher order pair. In case of manipulators all joints are restricted to one degree of freedom.

A.2 Kinematic Equations of Motion for Moving Platform

A skid steer type platform must satisfy non holonomic constraints and can't follow an arbitrary path through configuration space. Assume that the wheels do not slip, the

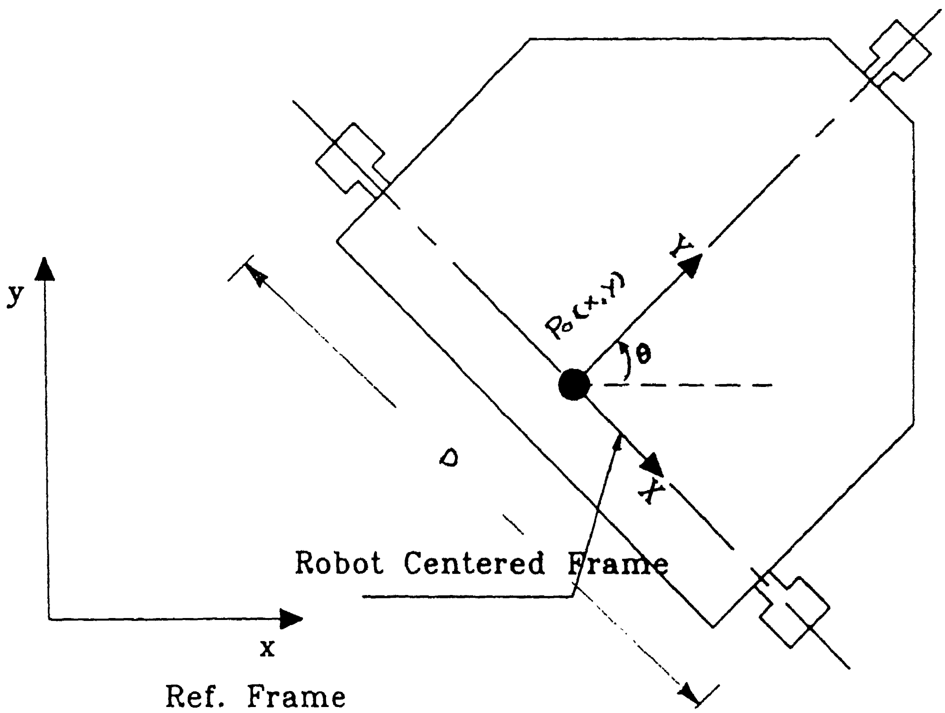


Fig. A.1 Modeling of the Moving Platform.

configuration of the platform is described by three variables; the cartesian coordinates x and y of the origin of the Robot centered frame (Robot centered frame is attached to the Moving Platform at the mid point of wheel axis and it moves along with the Moving Platform) with respect to a reference frame and the orientation, θ, of the Y axis of the robot centered frame with respect to reference frame x axis as shown in Figure A.1. The joint variables are the wheels translational displacements, denoted by x_R and x_L (representing the angular rotation times the radius of the right and left wheels respectively). V_R and V_L are the wheels translational velocities. ω_R and ω_L are the angular velocities of right and left wheels respectively of the platform. θ_R and θ_L are the angles of rotation of right and left wheels respectively of the platform.

Kinematic modeling links the cartesian variables to the control variables through the wheel velocities.

$$
\begin{aligned}
\dot{\theta} &= \frac{(V_R - V_L)}{D} && 1\\
\dot{x} &= \frac{(V_R + V_L)\cos(\theta)}{2} && 2\\
\dot{y} &= \frac{(V_R + V_L)\sin(\theta)}{2} && 3\\
\dot{y}\cos\theta - \dot{x}\sin\theta &= 0 && 4\\
\dot{x}_R &= V_R && 5\\
\dot{x}_L &= V_L && 6\\
\dot{\theta}_R &= \omega_R && 7\\
\dot{\theta}_L &= \omega_L && 8
\end{aligned}
$$

A non holonomic equality constraint is a non integrable equation involving the configuration parameters and their derivatives. Equation 4 shows a non holonomic constraint that the velocity of the Moving Platform must be in the direction of axis of symmetry (Y axis).

Appendix B

B.1 Hardware requirements for an image processing system

The typical image processing system works with standard T.V format signals as its input and output. So before a microcomputer can do any image processing, some means to capture, store, and display an image frame must be provided. Further an appropriate interface to allow computer access to the image storage memory must be available. Additional hardware can then be added to enhance the speed of certain image processing operations.

The monochrome video signal format used, in general, is RS-170 standard. This standard is a subset of National Television Vision Systems Committee colour specification containing only monochrome video information. When colour processing is involved three RS-170 signals are often used to carry the red, green, and blue information.

The RS-170 defines the video composite signal that carries the visual and synchronizing information necessary to reconstruct a transmitted video image. Under this format the video image frame is defined as containing 525 lines in a 4:3 (horizontal to vertical) aspect ratio. Digitizing a video line to 512 pixels with a brightness resolution of 8 bits per pixel will fully capture the information contained in the RS-170 signal line.

The image processing system's front end hardware must take in RS-170 video, massage it with various analog prepossessing circuitry and convert it to a stream of digital values using a high speed flash A/D converter. The input analog functions consist of a preamplifier to buffer the signal from the transmission line, a black level clamp to reference the A.C coupled video to a known D.C voltage and a sync separator to recover the synchronizing information for system timing and control. The A/D

converter takes this analog voltage wave form and converts it to a stream of 8-bit pixels for storage in image memory.

The image processors back end must accept pixel data from the image memory, convert it to an analog signal using a high speed D/A converter, and process it through an output amplifier. The output amplifier serves to mix back synchronizing information as well as to buffer the outgoing signal. The resulting RS-170 video can then be fed to a standard video monitor for display. Fig.(B.1) illustrates the front end and back end analog processor hardware needed by a digital image processing system.

Between the front end and back end analog processing H/W sits the most essential portion of the image processor, The Image memory. It is here that the digital image sits for processing by the host micro computer. In addition, the image memory must accept incoming video from A/D converter for storage and provide outgoing video data to the D/A converter for display.

B.2 Common processing operations

Processing of digital image data arrays may occur either through H/W or S/W. H/W methods are faster than S/W methods, but expensive when compared to S/W methods and hence are usually reserved for special purpose applications. Also H/W implementation has none of the flexibility inherent in S/W. In a typical image processing system, only certain simple processing operations are contained in H/W, with more complex processing limited to S/W implementation.

Image processing operations can be broken into three categories.

1. Image enhancement.
2. Image analysis.
3. Image coding.

Image enhancement improves the quality of the image based on the requirements of the user.Image analysis computes relevant information about an image. For instance a statistical break down of the qualities of an image (histograms, profile maps etc.) may be generated. Image coding reduces an image data in size, to minimize storage space, transmission bandwidth etc ...

Appendix C

C.1 Pulse Width Modulation

A basic switching regulator consists of four major components as shown in the figure C.1:

1. Voltage source V_{in}
2. Switch S_1
3. Pulse generator V_{pulse}
4. Filter F_1

Voltage source, V_{in}, may be any D.C supply. It must supply the required output power and losses associated with the switching regulator. It must have the capacity to supply sufficient dynamic range voltage for line and load variations.

Switch, S_1, is typically a transistor connected as a power switch and is operated in the saturated mode. The pulse generator output alternately turns the switch ON and OFF.

Pulse generator, V_{pulse},produces an asymmetric square wave as shown in figure C.2 varying in frequency modulation or pulse width modulation respectively. The duty cycle of the pulse wave form determines the relationship between the input and output voltage. Duty cycle is the ratio of ON time, T_{on}, to the period T of the pulse wave form. The output voltage, V_0, of the switching regulator is a function of duty cycle and the input voltage V_{in}.
$V_0 = V_{in} * (T_{on}/(T_{on} + T_{off}))$
where
T_{off} = OFF time of the pulse wave form.
T_{on} = ON time of the pulse wave form.

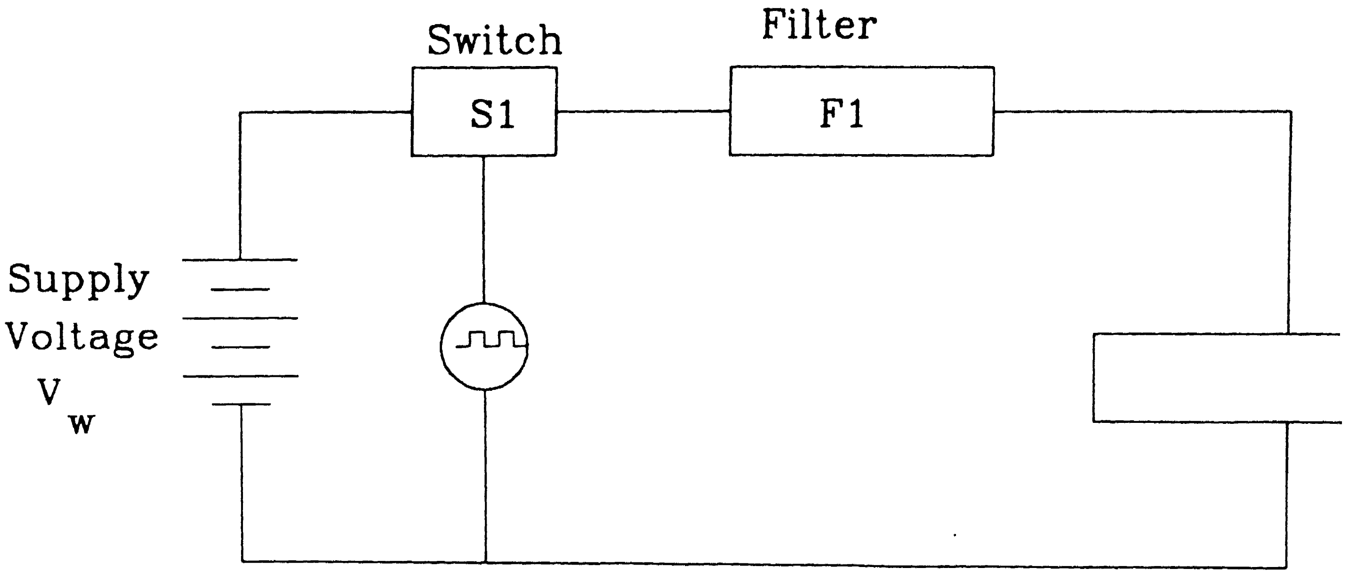
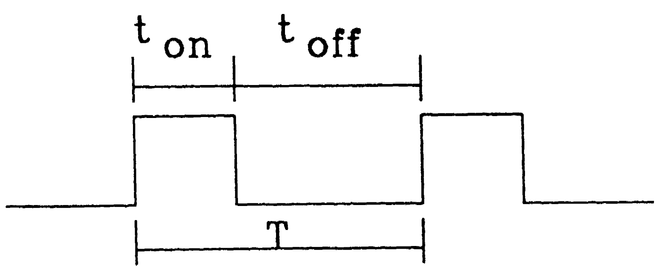


Figure C.1. Pulse width modulation



t_{on} = On time of the cyc

Figure C.2 Duty cycle

T = Time period

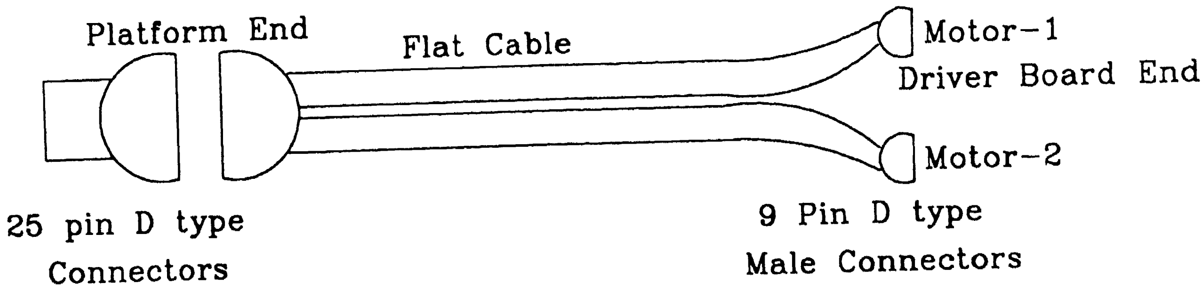
t_{off} = Off time of the cyc

If the time period T is constant, V_0 is directly proportional to the ON time, for a given value of V_{in}. This method of changing the output voltage by varying Tc referred to as Pulse Width Modulation. For the present work the pulses are gener through the printer port of the PC1.

Appendix D

Cable connections among various units of the work set up are below, in the form of tables. For serial communication between the personal computers a three wire connection has been made. The motors of the Platform are connected to the Driver board by means of a flat cable. A 25 pin D type connector has been used at the Platform end and two 9 pin D type connectors (for controlling the two motors independently) at the driver board end of the above mentioned cable. Another cable flat has been used for connecting Driver board and PC1 printer port. At driver board end an FRC connector is used and a 25 pin D type connector at PC1 printer port end.

Fig. D.1 Connections between Moving Platform and Driver.



25 pin D type female connector connections with the two D.C. motors of the Platform.

Pin No.	Signal Name
1	Motor-1 Power
2	Motor-1 Power
3	Encoder-1 Power + ve
4	Encoder-1 Power − ve
5	Encoder-1 signal (Blue Wire)
6	Encoder-1 signal (White Wire)
7	Motor-1&2 Ground (Green Wire)
8	Encoder-2 signal (White Wire)
9	Encoder-2 signal (Blue Wire)
10	Encoder-2 Power − ve
11	Encoder-2 Power + ve
12	Motor-2 Power
13	Motor-2 Power
14 to 25	Connected to Ground

9 pin D type male connector connections with the Driver board (Motor-1).

Pin No.	Cable No.
1	1 (Motor-1 Power + ve)
3	5 (Encoder-1 Power + ve)
4	- (Micro Switch)
5	7 (Logic Ground)
6	11 (Encoder-1 signal, White Wire)
8	9 (Encoder-1 signal, Blue Wire)
9	3 (Motor-1 Power − ve)

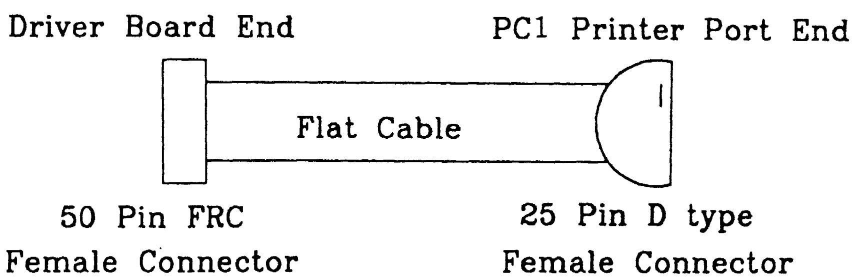
9 pin D type male connector connections with the Driver board (Motor-2).

Pin No.	Cable No.
1	25 (Motor-2 Power)
3	21 (Encoder-2 Power + ve)
4	- (Micro Switch)
5	19 (Logic Ground)
6	15 (Encoder-2 signal, White Wire)
8	17 (Encoder-2 signal, Blue Wire)
9	23 (Motor-2 Power)

25 pin D type male connector connections with flat cable (Platform end).

Pin No.	Cable No.	Pin No.	Cable No.
1	1	14	2
2	3	15	4
3	5	16	6
4	7	17	8
5	9	18	10
6	11	19	12
7	13	20	14
8	15	21	16
9	17	22	18
10	19	23	20
11	21	24	22
12	23	25	24
13	25		

Fig. D.2 Connections between Driver and PC1 printer port.



50 pin FRC female connector connections with the Driver board.

50 Pin FRC Female	Signal Name
1	Protect
3	AFOR
11	AREV
19	BFOR
27	BREV
43	AOPT1
45	AOPT2
47	BOPT1
49	BOPT2

25 pin D type female connector, at PC1 printer port end, with 50 pin FRC female connector, at Driver board end, by a flat cable.

25 Pin D Type Female	50 Pin FRC Female
1	1
2	3
3	11
4	19
5	27
15	43
13	45
12	47
10	49

PC1 printer port connections with 25 pin D type female connector.

Pin No.	Signal Name on Driver board	Printer port Data bit No.	Printer port Pin No.	Printer port Address
1	Protection	5	1	0x27A
2	A-Forward	0	2	0x278
3	A-Reverse	1	3	0x278
4	B-Forward	2	4	0x278
5	B-Reverse	3	5	0x278
10	BOPT2	6	10	0x279
12	BOPT1	5	12	0x279
13	AOPT2	4	13	0x279
15	AOPT1	3	15	0x279

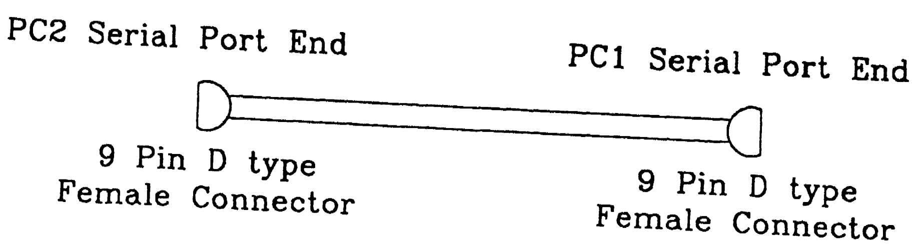


Fig. D.3 Serial port communication connections between PC1 and PC2.

9 pin D type female connector connections with PC1 RS-232 serial port.

Pin No.	Cable No.
1	1
2	2
3	3

9 pin D type female connector connections with PC2 RS-232 serial port.

Pin No.	cable No.
1	1
2	3
3	2

References

1. S.Ishikawa et al., " A method of Image Guided Vehicle using white line recognition ", *Proc. IEEE Conf. Computer Vision and Pattern Recognition* June 1986, pp. 47-53.

2. E.S. McVey et al., " Range Measurements by a Mobile Robot using a navigation line ", *IEEE Trans. Pattern Analy. Machine Intelligence*, Vol.PAMI-8, no.1,pp.105-109, Jan 1986.

3. M.Herbert and T.Kanade, " Outdoor Scene analysis using range data ", *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, Apr 1986.

4. S.Ishikawa et al., " Visual navigation of an autonomous vehicle using white line recognition ", *IEEE Trans. Pattern analysis and Machine Intelligence*, Vol.10,No.5,Sep.1988.

5. J.Y. Zheng, M.Barth and T.Tsuji, " Autonomous landmark selection for route recognition by a mobile robot ", *Proc. IEEE International Conf. on Robotics and Automation*, 1991.

6. H.P.Moravec, " Robot Rover visual navigation ", *UMI research press*, 1981.

7. H.P.Moravec, " The Stanford Cart and CMU Rover ", *Proc of IEEE on Robotics and Automation*, 1983.

8. G.Giralt, R.Sobek and R.Chatila, " A multi level planning and navigation system for a mobile robot: A first approach to Hilare ", *Proc. sixth Inter joint Conf. on A.I* , 1979.

9. R.Chatila and J.P.Laumond, " Position referencing and consistent world modeling for mobile robots ", *Proc. IEEE Conf. on Rob. Automation*, 1985.

10. D.J.Kriegman, E.Triendl and T.O.Binford, " Stereo vision and navigation in buildings for mobile robots ", *IEEE Trans. Rob. and Automation*,1989.

11. N.Ayache and Faugers, " Maintaining representations of the environment of a mobile robot ", *IEEE Trans on Rob. and Automation*,1989.

12. Kosaka and Kak, " Fast vision guided mobile robot navigation ", *Image understanding*, 1992.

13. J.Latombe, " Robot Motion Planning ", Kluwer, Boston,1991.

14. T.Lozano Perez and M.K.Wesley, " An algorithm for planning collision free paths among polyhedral obstacles ", *Commun. ACM22(10)* , 1979.

15. O.Khatib, " Real time obstacle avoidance for manipulators and mobile robots ", *Int. J.Robotics Res.*,1986.

16. Kanayama.Y and Hartman B.I, " Smooth local path planning for autonomous vehicles ", *Proc. IEEE Inter. Conf. on Rob. and Automation*, 1989.

17. P.F.Muir and C.P.Neuman, " Kinematic Modeling of wheeled mobile robots ", *Journal of Robotic systems* , Vol.4,1987.

18. E.D Dickmanns and V.Graefe, " Dynamic Monocular machine vision and applications ", *Machine vision and applications* , Vol.1, 1988.

19. J.K.Aggarwal and R.O. Duda, " Computer analysis of moving polygonal images ", *IEEE Trans. Compu.*, 1975.

20. W.N.Martin and J.K.Aggarwal, " Computer analysis of dynamic scenes containing curvilinear figures ", *Pattern Recognition-2*, 1979.

21. J.L.Potter, " Scene segmentation using motion information" , *Comp. Graphics Image Proc.*, 1977.

22. W.K.Chow and J.K.Aggarwal, " Comp. analysis of planar curvilinear moving images ", *IEEE Trans. on Comp.*, 1977.

23. A.Mitiche ,O.Fangeras and J.K. Aggarwal, " Counting straight lines ", *Comp. Vision, Graphics and Image Processing*, 1989.

24. A.Mitiche and G.Habelrih, " Interpretation of straight line correspondence using angular relations ", *Pattern Recognition*, 1989.

25. C.H.Lee, " Interpreting Image Curve from multi frame ", *Artificial Intelligence*,1988.

26. Nilanjan Sarkar, Xiaoping Yun, and Vinay Kumar, " Control of Mechanica systems with rolling constraints: Application to Dynamic Control of Mobil Robots ", *The International J. Robotics Research*, 1994.

27. David B. Reister and Francois G.Pin, " Time Optimal Trajectories for Mo bile robots with two independently Driven Wheels", *The International J Robotics Research*, 1994.